7-2018

Incremental and Adaptive L1-Norm Principal Component Analysis: Novel Algorithms and Applications

Mayur Dhanaraj
mxd6023@rit.edu

Incremental and Adaptive L1-Norm Principal Component Analysis: Novel Algorithms and Applications

by

Mayur Dhanaraj

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science in Electrical Engineering

Supervised by

Dr. Panos P. Markopoulos
Department of Electrical and Microelectronic Engineering
Kate Gleason College of Engineering
Rochester Institute of Technology, Rochester, New York
July 2018

Approved by:

Dr. Panos P. Markopoulos, Assistant Professor
Thesis Advisor, Department of Electrical and Microelectronic Engineering

---

Dr. Andreas Savakis, Professor
Committee Member, Department of Computer Engineering

Dr. Sohail A. Dianat, Professor
Committee Member, Department of Electrical and Microelectronic Engineering

---

Dr. Sohail A. Dianat, Professor
Department Head, Department of Electrical and Microelectronic Engineering

Thesis Release Permission Form

Rochester Institute of Technology
Kate Gleason College of Engineering

Title:

Incremental and Adaptive L1-Norm Principal Component Analysis: Novel Algorithms and Applications

I, Mayur Dhanaraj, hereby grant permission to the Wallace Memorial Library to reproduce my thesis in whole or part.

Mayur Dhanaraj

Date

Dedication

I dedicate this work to my mother Dr. Gnaneswari Gopal, my sisters Mrunalini Rout and Mrudula Dhanraj, my grandfather K. Gopal Naidu and my best friends Prakruthi Manjunath, Prasidh P Kumar and Akash M Bushan for their continued love and support. I also dedicate this work to my late father Dhanaraj Guntoor Muniswamy, my late paternal aunt Amruthamma M and my late maternal uncle Sumanth Kumar for their blessings. Last but not the least I dedicate this work to the almighty for giving me the strength and ability to complete this work.

Acknowledgments

There are a number of people without whom this thesis might not have been possible, and to whom I am forever indebted.

To my mother, Dr. Gnaneswari Gopal for giving birth to me and nurturing me, for being a pillar of my family and for her constant support and advice. I would not be who I am today without her support. A very special thanks to you mother.

To my advisor and mentor, Dr. Panos P. Markopoulos for being an excellent guide, a patient listener, an inspiration and an extraordinary teacher. Thank you professor.

I thank each member of my family for being very supportive to me throughout my life.

I thank Mr. Mahesh Narayana for constantly encouraging me with his words of wisdom.

I thank my fellow lab partners Dimitis Chachlakis and Ruslan Dautov (both PhD students), for their invaluable advice whenever needed.

I thank Dr. Sailaja Viswanath and Dr. T. M. Manjunath for their invaluable support to my mother in times of need.

I also thank my thesis committee members Dr. Andreas Savakis and Dr. Sohail A. Dianat for taking their valuable time to review my thesis and for their invaluable suggestions that led to the betterment of this thesis.

Abstract

Incremental and Adaptive L1-Norm Principal Component Analysis: Novel Algorithms and Applications

Mayur Dhanaraj

Supervising Professor: Dr. Panos P. Markopoulos

L1-norm Principal-Component Analysis (L1-PCA) is known to attain remarkable resistance against faulty/corrupted points among the processed data. However, computing L1-PCA of "big data" with large number of measurements and/or dimensions may be computationally impractical. This work proposes new algorithmic solutions for incremental and adaptive L1-PCA. The first algorithm computes L1-PCA incrementally, processing one measurement at a time, with very low computational and memory requirements; thus, it is appropriate for big data and big streaming data applications. The second algorithm combines the merits of the first one with additional ability to track changes in the nominal signal subspace by revising the computed L1-PCA as new measurements arrive, demonstrating both robustness against outliers and adaptivity to signal-subspace changes. The proposed algorithms are evaluated in an array of experimental studies on subspace estimation, video surveillance (foreground/background separation), image conditioning, and direction-of-arrival (DoA) estimation.

List of Contributions

- Novel Algorithm for Incremental L1-norm Principal-Component Analysis.
- Novel Algorithm for Adaptive L1-norm Principal-Component Analysis.
- Numerical Studies on Outlier-Resistant Signal Subspace Estimation and Tracking.
- Experimental Studies on Image Conditioning, Specifically Glare/Shadow Artifacts Removal from Face Images.
- Experimental Studies on Video Background/Foreground Extraction.
- Experimental Studies on Jammer-Resistant Direction-of-Arrival (DoA) Estimation and Tracking.
- **Publications:**

  1) P. P. Markopoulos, M. Dhanaraj and A. Savakis, "Algorithms for incremental and adaptive L1-norm principal-component analysis", *IEEE J. Sel. Top. Signal Process* (under revision).

  2) M. Dhanaraj and P. P. Markopoulos, "Novel algorithm for incremental L1-norm principal-component analysis", *IEEE/EURASIP Euro. Signal Process. Conf. (EUSIPCO 2018)*, Rome, Italy, Sept. 2018 (to appear).

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Principal-Component Analysis

Principal-Component Analysis (PCA) [1–3] is a cornerstone of data analysis that strives to extract the most important low-rank component of a multivariate dataset. Formally, PCA seeks to maximize the total variance of the projection of all original data onto a small number of orthogonal directions (principal components) that define a lower dimensional subspace. Over the past decades, PCA has found numerous applications in, e.g., signal processing [4, 5], wireless communications [6, 7], machine learning [8, 9], pattern recognition [10], video/image processing [11], and biomedical signal processing [12–15].

In its standard formulation, PCA approximates data matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N] \in \mathbb{R}^{D\times N}$ by another low-rank matrix product $\mathbf{QS}^T$, where $\mathbf{Q} \in \mathbb{R}^{D\times K}$, $\mathbf{S} \in \mathbb{R}^{N\times K}$ and $K < d = \text{rank}(\mathbf{X})$, so that the squared L2-norm of the approximation error is minimized. That is, standard PCA is formulated as [16]

$$(\mathbf{Q}_{L2}, \mathbf{S}_{L2}) = \underset{\mathbf{Q}\in\mathbb{R}^{D\times K},\ \mathbf{S}\in\mathbb{R}^{N\times K}}{\operatorname{argmin}} \|\mathbf{X} - \mathbf{QS}^T\|_F^2, \tag{1.1}$$

where the L2-norm (or Frobenius norm) $\|\cdot\|_F^2$ returns the sum of squared entries of its matrix argument. Observing that for any given $\mathbf{Q}$, $\mathbf{S} = \mathbf{X}^T\mathbf{Q}$ minimizes the error in (1.1),

we obtain the following formulation

$$\mathbf{Q}_{L2} = \underset{\substack{\mathbf{Q}\in\mathbb{R}^{D\times K}\\ \mathbf{Q}^T\mathbf{Q}=\mathbf{I}_K}}{\operatorname{argmin}} \|\mathbf{X} - \mathbf{Q}\mathbf{Q}^T\mathbf{X}\|_F^2. \tag{1.2}$$

(1.2) is known as the L2 error minimization problem and can be equivalently rewritten as

$$\mathbf{Q}_{L2} = \underset{\substack{\mathbf{Q}\in\mathbb{R}^{D\times K}\\ \mathbf{Q}^T\mathbf{Q}=\mathbf{I}_K}}{\operatorname{argmin}} \sum_{i=1}^{N} \|\mathbf{x}_i - \mathbf{Q}\mathbf{Q}^T\mathbf{x}_i\|_F^2,$$

where $\mathbf{x}_i = [\mathbf{X}]_{:,i}$ is the i-th sample of $\mathbf{X} \in \mathbb{R}^{D\times N}$. I.e., (1.2) aims at finding the $\mathbf{Q}$ that minimizes sum of the squared Frobenius norm error of each entry of the data matrix $\mathbf{X}$ and its projection onto $\mathbf{Q}$ where $\mathbf{Q} \in \mathbb{R}^{D\times K}$ and $\mathbf{Q}^T\mathbf{Q} = \mathbf{I}_K$, depicted in Figure 1.1.

We know that the squared Frobenius norm of a matrix $\mathbf{M}$ is equal to the trace of the product of the transposed matrix with itself, i.e., $\|\mathbf{M}\|_F^2 = tr(\mathbf{M}^T\mathbf{M})$, where $tr(\cdot)$ returns the sum of diagonal entries of its matrix argument. Therefore, (1.2) can be expanded as

$$\begin{aligned}
\mathbf{Q}_{L2} = \underset{\substack{\mathbf{Q}\in\mathbb{R}^{D\times K}\\ \mathbf{Q}^T\mathbf{Q}=\mathbf{I}_K}}{\operatorname{argmin}} \|\mathbf{X} - \mathbf{Q}\mathbf{Q}^T\mathbf{X}\|_F^2 &= \underset{\substack{\mathbf{Q}\in\mathbb{R}^{D\times K}\\ \mathbf{Q}^T\mathbf{Q}=\mathbf{I}_K}}{\operatorname{argmin}} tr\left[(\mathbf{X} - \mathbf{Q}\mathbf{Q}^\mathbf{T}\mathbf{X})^T(\mathbf{X} - \mathbf{Q}\mathbf{Q}^\mathbf{T}\mathbf{X})\right] \\
&= \underset{\substack{\mathbf{Q}\in\mathbb{R}^{D\times K}\\ \mathbf{Q}^T\mathbf{Q}=\mathbf{I}_K}}{\operatorname{argmin}} \left[\|\mathbf{X}\|_F^2 - \|\mathbf{Q}^T\mathbf{X}\|_F^2\right] \\
&= \underset{\substack{\mathbf{Q}\in\mathbb{R}^{D\times K}\\ \mathbf{Q}^T\mathbf{Q}=\mathbf{I}_K}}{\operatorname{argmax}} \|\mathbf{Q}^T\mathbf{X}\|_F^2
\end{aligned}$$

That is, $\mathbf{Q}_{L2}$ can equivalently be found by the projection maximization

$$\mathbf{Q}_{L2} = \underset{\substack{\mathbf{Q}\in\mathbb{R}^{D\times K}\\ \mathbf{Q}^T\mathbf{Q}=\mathbf{I}_K}}{\operatorname{argmax}} \left\|\mathbf{Q}^T\mathbf{X}\right\|_F^2 \tag{1.3}$$

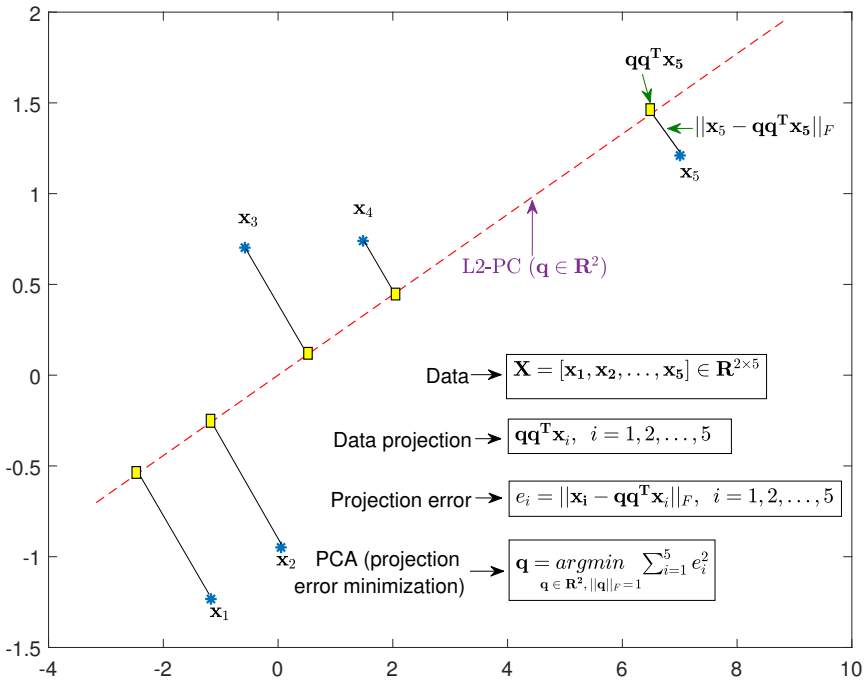


Figure 1.1: Projection error minimization PCA.

and, accordingly, $\mathbf{S}_{L2} = \mathbf{X}^T\mathbf{Q}_{L2}$. Again, (1.3) can be rewritten as

$$\mathbf{Q}_{L2} = \underset{\substack{\mathbf{Q}\in\mathbb{R}^{D\times K}\\ \mathbf{Q}^T\mathbf{Q}=\mathbf{I}_K}}{\operatorname{argmax}} \sum_{i=1}^{N} \left\|\mathbf{Q}^T\mathbf{x}_i\right\|_F^2$$

. I.e., (1.3) aims at finding the $\mathbf{Q}$ that maximizes the sum of squared Frobenius norm of projection magnitude of each entry of the data matrix $\mathbf{X}$ onto $\mathbf{Q}$ and is depicted in Figure 1.2.

The solution to (1.1), (1.2) and (1.3), $\mathbf{Q}_{L2}$, contains the K-dominant left singular-vectors of $\mathbf{X}$, obtained through standard Singular-Value Decomposition (SVD) [17]. Therefore, standard PCA is both conceptually simple and computationally efficient, with cost $O(NDmin(N, D))$.

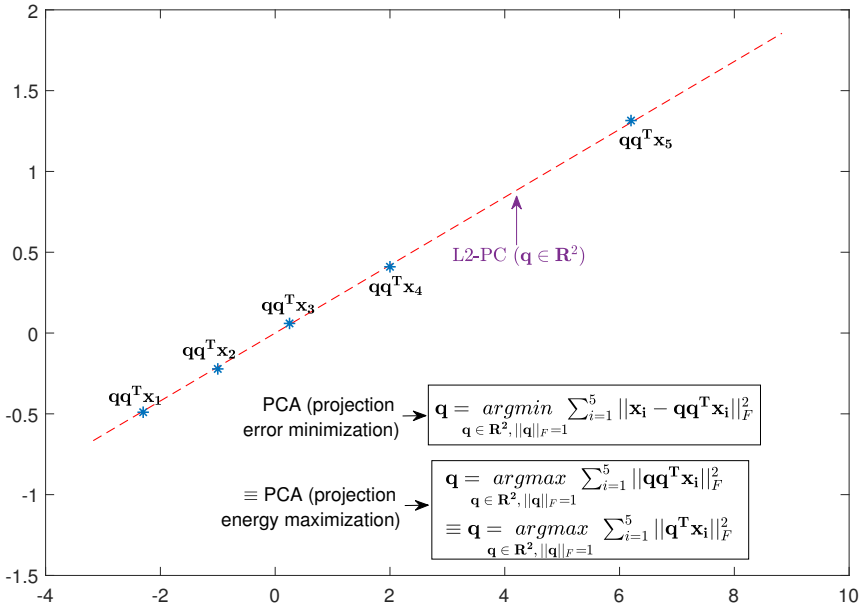


Figure 1.2: Projection variance maximization PCA.

1.2 Outliers

In the big data era, real-world datasets often contain irregular or corrupted measurements that lie far from the nominal data. Such measurements are commonly referred to as "outliers" [18] and may appear due to various causes such as intermittent sensor malfunctions, errors in data transcription, transmission or labeling, deliberate jamming, or sporadic environmental changes among others. A brief summary of few applications of PCA and possible causes of outliers in various domains is provided in Table 1.1.

Regretfully, standard PCA is known to be very fragile in the presence of outliers, even if they appear in a small fraction of the processed data. The reason is that the L2-norm objective of PCA in (1.3), $\|\mathbf{Q}^T\mathbf{X}\|_F^2 = \sum_{i=1}^{N} \|\mathbf{Q}^T\mathbf{x}_i\|_2^2$, places squared emphasis on each data point,

DOMAIN	SUB-DOMAIN	APPLICATION	POSSIBLE CAUSE OF OUTLIERS
Computer vision and image processing	Image processing	Glare and/or shadow specularity removal	Varying illumination
		Face recognition	Error in data labeling and occlusions
	Video processing	Background/foreground extraction; used in security surveillance, object recognition, event detection, etc.	Foreground movement, occlusions and varying illumination
		Visual tracking; used in target localization, eye tracking for disease diagnosis, etc.	Occlusions and varying illumination
Signal processing	RADAR signal processing	Direction of arrival (DoA) estimation and tracking; used in tracking enemy aircraft, estimating the direction of signal of interest in wireless communication, etc.	Deliberate jamming and sporadic environmental changes
Data analysis	Web-data analysis	Recommender systems; used by e-shopping sites and online streaming services like Amazon and Netflix to recommend products	Malicious or deliberate tampering and data mislabeling
	Bioinformatics	DNA sequence analysis and genome annotation; used in medicine to predict genetic disease and in forensics for forensic identification and paternity testing	Intermittent sensor malfunction and labelling/transcription error
Machine learning	Classification	Disease diagnosis and image/text classification	Error in data labeling and sensor malfunction
	Dimensionality reduction	Data visualization and cure to curse of dimensionality	Acquisition error, sensor malfunction and transcription/transmission error

Table 1.1: Possible causes of outliers in few applications of interest.

therefore benefiting peripheral, outlying points. A simple line-fitting experiment demonstrates the outlier-sensitivity of L2-PCA in Figure 1.3 and Figure 1.4. In Figure 1.3, there are no outliers and hence the L2 principal-component (L2-PC) is very close to the maximum variance line, however in Figure 1.4, there are 2 outliers among the processed data and therefore the L2-PC deviates away from the maximum variance line (depicting clear attraction towards outlying points). Therefore, the use of traditional PCA in real-world

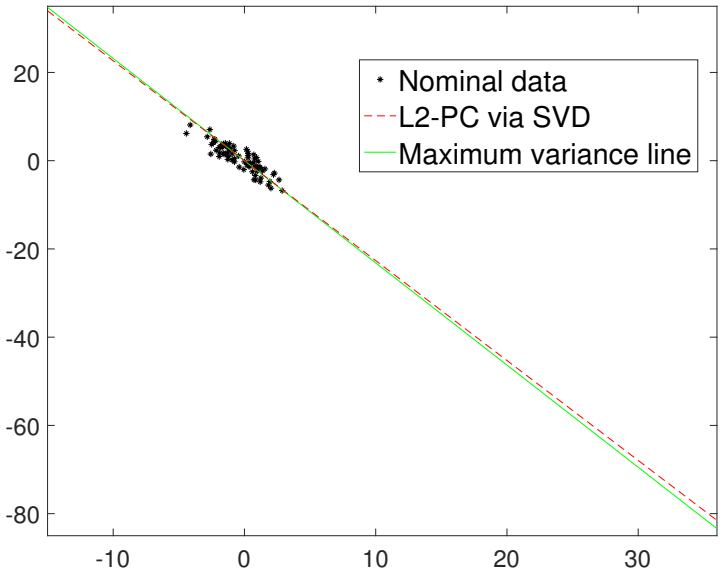


Figure 1.3: Line-fitting for nominal data.

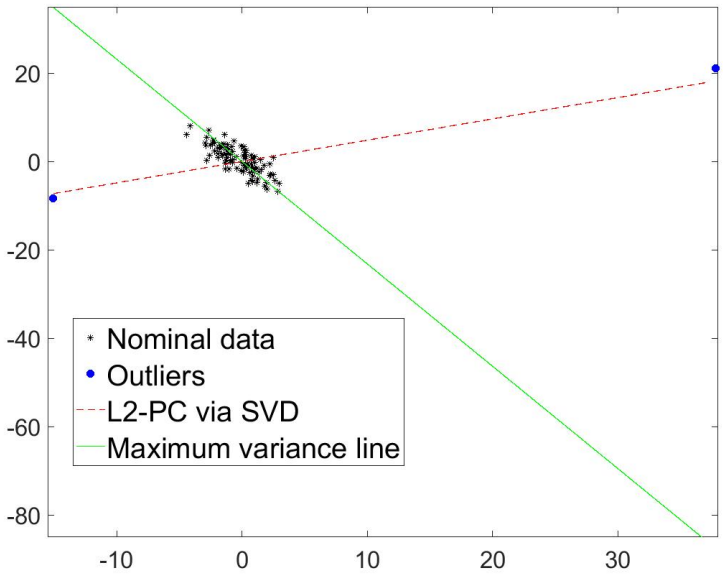


Figure 1.4: Line-fitting for outlier-corrupted data.

and/or big-data setting where outliers are common leads to unreliable solutions, creating a need for robust PCA.

The contribution of this thesis are as follows:

- Novel Algorithm for Incremental L1-norm Principal-Component Analysis.
- Novel Algorithm for Adaptive L1-norm Principal-Component Analysis.
- Numerical Studies on Outlier-Resistant Signal Subspace Estimation and Tracking.
- Experimental Studies on Image Conditioning, Specifically Glare/Shadow Artifacts Removal from Face Images.
- Experimental Studies on Video Background/Foreground Extraction.
- Experimental Studies on Jammer-Resistant Direction-of-Arrival (DoA) Estimation and Tracking.

Chapter 2

Background Review

2.1 Robust Principal-Component Analysis

To counteract the impact of outliers in data analysis and processing, researchers have long focused on developing "robust" subspace estimation alternatives. In the popular Robust PCA (RPCA) line of research, the outlier-corrupted dataset is modeled as the summation of a low-rank component that describes the nominal subspace and a sparse component that captures the outliers [19–24]. At first, this RPCA problem formulation suggests a solution where we seek to find the least-ranked low rank component (that best describes the nominal subspace) and the most sparse component to model any outliers amongst processed data. Mathematically, given $\mathbf{X} = \mathbf{L} + \mathbf{S} + \mathbf{n}$ and λ, a constant, where $\mathbf{L}$ and $\mathbf{S}$ are unknowns, $\mathbf{L}$ being the sought-after low rank component, $\mathbf{S}$ being the sparse component, and $\mathbf{n}$ – the noise in the data matrix $\mathbf{X}$, the RPCA optimization problem can be formulated as

$$(\mathbf{L}, \mathbf{S}) = \underset{\substack{\mathbf{L},\,\mathbf{S} \\ \mathbf{L}+\mathbf{S}=\mathbf{X}}}{\operatorname{argmin}} \ \operatorname{rank}(\mathbf{L}) + \lambda ||\mathbf{S}||_0, \tag{2.1}$$

where the zero-norm $||\cdot||_0$ returns the number of non-zero entries in its matrix argument. (2.1) is a non-convex problem that is NP-hard and no efficient solution exists in literature.

However, (2.1) can be reformulated as a tractable convex optimization problem by replacing the rank with nuclear norn $||\mathbf{L}||_*$ and L0-norm by L1-norm, i.e.,

$$(\mathbf{L}, \mathbf{S}) = \underset{\substack{\mathbf{L},\ \mathbf{S} \\ \mathbf{L}+\mathbf{S}=\mathbf{X}}}{\operatorname{argmin}} \ ||\mathbf{L}||_* + \lambda||\mathbf{S}||_1, \tag{2.2}$$

where nuclear-norm $||\cdot||_*$ returns the sum of singular values of its matrix argument and L1-norm $||\cdot||_1$ returns the sum of absolute entries of its matrix argument. It was shown in [25] that (2.2) is indeed a convex optimization problem and provided enough conditions to prove the same. Therefore, (2.2) can be solved using convex optimization techniques or algorithms proposed in [19–24]. Once $\mathbf{L}$ and $\mathbf{S}$ are successfully extracted, what remains of $\mathbf{X}$ is the noise $\mathbf{n}$ and can be neglected.

Another outlier-resistant minimum rank solution is proposed in [26] by solving a convex optimization problem, namely nuclear-norm minimization. Authors in [27] propose an efficient rotational invariant L1-norm PCA (R1-PCA) to perform robust PCA. Robust subspace learning (RSL) in [28, 29] proposes algorithms that detect outliers and replace them by neighboring nominal points, or places a weight on each data point to downgrade outliers among the processed data. Fast and low complexity algorithms for robust PCA via gradient descent are proposed in [30].

In another line of research, PCA is robustified by substituting the L2-norm in 1.1 by the L1-norm [31–33]. To date, no exact solution exists for this L1-norm error-minimization PCA, for general $K \geq 1$. Another popular approach substitutes the L2-norm by the L1-norm directly on (1.3), effectively removing the squared emphasis that standard PCA places on each datum. This L1-*projection-maximization* approach is also known as L1-PCA, and

is discussed in the next subsection.

2.2 L1-norm Principal-Component Analysis

L1-PCA [34–36] is another robust approach that performs outlier-resistant PCA and it is mathematically formulated as

$$\mathbf{Q}_{L1} = \underset{\substack{\mathbf{Q}\in\mathbb{R}^{D\times K} \\ \mathbf{Q}^T\mathbf{Q}=\mathbf{I}_K}}{\text{argmax}} \ ||\mathbf{Q}^T\mathbf{X}||_1, \tag{2.3}$$

where L1-norm $\|\cdot\|_1$ returns the sum of absolute entries of its matrix argument. Contrary to what is true for standard PCA in (1.1)-(1.3), L1-projection-maximization PCA and L1-error-minimization PCA are not equivalent. Moreover, it has been shown [34] that the K L1-PCs in (2.3) have to be jointly computed.

In [36], Kwak proposed an early approximate solver for (2.3) with complexity $O(N^2DK)$. The solver of [36] first approximates the dominant L1-PC ($K = 1$) of $\mathbf{X}$ and then computes the remaining $K - 1$ L1-PCs through a sequence of deflating null-space projections. In [37], Nie et al. targeted the problem of computing jointly all $K \geq 1$ L1-PCs of $\mathbf{X}$ and, for this task, they proposed a "non-greedy" alternating-optimization algorithm of complexity $O(N^2DK + NK^3)$. A semi-definite programming (SDP) approach for (2.3) was proposed by McCoy and Tropp in [38] with cost $O(KN^{3.5}log(1/\epsilon) + KL(N^2 + DN))$ for desired accuracy ϵ. Authors in [39] presented a low-cost/high-performance L1-PCA/SVD hybrid model.

The exact solution to L1-PCA was delivered for the first time in [34] where authors reformulated (2.3) as an equivalent combinatorial optimization problem over NK $\{\pm 1\}$

variables. The algorithms of [34] solve (2.3) exactly with complexity $O(2^{NK})$, in general, or $O(N^{dK-K+1})$ when $d = \text{rank}(\mathbf{X})$ is a constant with respect to N.

2.2.1 Exact Solution

The authors in [34] showed that if

$$\mathbf{B}_{\text{opt}} = \underset{\mathbf{B} \in \{\pm 1\}^{N \times K}}{\text{argmax}} \; ||\mathbf{X}\mathbf{B}||_*, \tag{2.4}$$

where nuclear norm $\| \cdot \|_*$ returns the sum of the singular values of its matrix argument, then L1-PCA in (2.3) is solved by

$$\mathbf{Q}_{L1} = \Phi(\mathbf{X}\mathbf{B}_{\text{opt}}) \tag{2.5}$$

where, for any tall matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ with SVD $\mathbf{A} \overset{\text{SVD}}{=} \mathbf{U}\mathbf{\Sigma}_{n \times n}\mathbf{V}^T$, $\Phi(\mathbf{A}) = \mathbf{U}\mathbf{V}^T$. In addition, [34] showed that $||\mathbf{X}^T\mathbf{Q}_{L1}||_1 = ||\mathbf{X}\mathbf{B}_{\text{opt}}||_*$ and

$$\mathbf{B}_{\text{opt}} = \text{sgn}\left(\mathbf{X}^T\mathbf{Q}_{L1}\right). \tag{2.6}$$

Therefore L1-PCA in (2.3) can be cast as an equivalent combinatorial optimization problem over antipodal binary variables in $\{\pm 1\}$. The first optimal algorithm in [34] performs exhaustive search over the entire feasibility set of (2.4), $\{\pm 1\}^{N \times K}$, to obtain a solution $\mathbf{B}_{\text{opt}}$ with exponential complexity $O(2^{NK})$. The second optimal algorithm in [34] first constructs a polynomial-size subset of $\{\pm 1\}^{N \times K}$, $\mathcal{B}$, wherein a solution to (2.4) is proven to exist, then it searches exhaustively among the elements $\mathcal{B}$ to obtain $\mathbf{B}_{\text{opt}}$ and, through an additional SVD step in (2.5), returns the solution to L1-PCA in (2.3) with overall polynomial cost

$O(N^{dK-K+1})$. It is noticeable that the cost of both exact L1-PCA solvers may be impractical in big data applications (large N and/or d), and thus there was a need for algorithms that perform L1-PCA efficiently at lower computation cost while retaining the robustness of L1-norm against outliers.

To this end, Markopoulos et al. in [35] introduced a bit-flipping-based approximate solver for (2.3), labeled *L1-BF*, with cost $O(NDmin\{N, D\} + N^2(K^4 + DK^2) + NDK^3)$, and showed that L1-BF attains very low (if any) performance degradation in the L1-PCA metric, often outperforming its counterparts.

2.2.2 Efficient L1-PCA Through Bit-Flipping (L1-BF)

L1-BF is a state-of-the-art efficient algorithm for L1-PCA based on optimal single bit-flipping iterations [35]. L1-BF has similar cost with standard PCA (i.e., SVD), it exhibits sturdy outlier resistance, and it appears to outperform most of its counterparts in the L1-PCA metric. The incremental and adaptive L1-PCA calculators presented in this work are motivated by L1-BF, which is concisely presented below.

L1-BF commences at some initialization $\mathbf{B}(0) \in \{\pm 1\}^{N \times K}$ (arbitrary or better – a more intelligent (*sv-sign*) initialization as shown in [35] for faster convergence) and executes a sequence of optimal single-bit-flipping iterations across which the metric in (2.4) monotonically increases. Specifically, at each iteration, L1-BF examines all bits and recognizes the single bit which, when flipped, will offer the highest increase to the metric of (2.4). That is, at the t-th iteration, L1-BF finds

$$(n, k) = \underset{\substack{(m,l) \in \{1,2,\ldots,N\} \\ \times \{1,2,\ldots,K\}}}{\operatorname{argmax}} \left\| \mathbf{X}\mathbf{B}(t) - 2\left[\mathbf{B}(t)\right]_{m,l} \mathbf{x}_m \mathbf{e}_{l,K}^T \right\|_*, \tag{2.7}$$

Algorithm L1-BF

Input: $\mathbf{X} \in \mathbb{R}^{D \times N}$, init. $\mathbf{B} \in \{\pm 1\}^{N \times K}$, $K \leq \text{rank}(\mathbf{X})$,

$\mathbf{B} \leftarrow \texttt{BF}(\mathbf{X}, \mathbf{B}, K)$

$(\mathbf{U}, \mathbf{\Sigma}_{K \times K}, \mathbf{V}) \leftarrow \texttt{SVD}(\mathbf{XB})$

$\mathbf{Q} \leftarrow \mathbf{U}\mathbf{V}^T$

Output: $(\hat{\mathbf{B}}, \hat{\mathbf{Q}}) \leftarrow (\mathbf{B}, \mathbf{Q})$

Function: $\mathbf{B} \leftarrow \texttt{BF}(\mathbf{X}_{D \times N}, \mathbf{B}, K)$

$\omega \leftarrow K||\mathbf{X}[\mathbf{B}]_{:,1}||_2$

while true (or terminate at NK iterations)

  for $m \in \{1, 2, \ldots, N\}$, $l \in \{1, 2, \ldots, K\}$

    $a_{m,l} \leftarrow ||\mathbf{XB} - 2[\mathbf{B}]_{m,l}\mathbf{x}_m\mathbf{e}_{l,K}^T||_*$

  $(n, k) \leftarrow \text{argmax}_{m,l}\ a_{m,l}$

  if $\omega < a_{n,k}$

    $[\mathbf{B}]_{n,k} \leftarrow -[\mathbf{B}]_{n,k}$, $\omega \leftarrow a_{n,k}$

  else, break

Return $\mathbf{B}$

Figure 2.1: Pseudocode of L1-BF algorithm.

where $\mathbf{e}_{l,K}$ denotes the l-th column of the size-K identity matrix $\mathbf{I}_K$ and $\mathbf{x}_m$ is the m-th column of data matrix $\mathbf{X}$. Thereafter, L1-BF flips the (n, k)-th bit of $\mathbf{B}(t)$ setting

$$\mathbf{B}(t+1) = \mathbf{B}(t) - 2[\mathbf{B}(t)]_{n,k}\mathbf{e}_{n,N}\mathbf{e}_{k,K}^T. \tag{2.8}$$

Bit-flipping terminates at iteration t if the nuclear norm in (2.4) cannot further increase by any single-bit flip. Upon termination L1-BF returns $\hat{\mathbf{B}} = \mathbf{B}(t)$ as an approximation to $\mathbf{B}_{\text{opt}}$ in (2.4) and $\hat{\mathbf{Q}} = \Phi(\mathbf{XB}(t))$ as an approximation to $\mathbf{Q}_{L1}$ in (2.3), in accordance with (2.5). It was shown in [35] that the bit-flipping iterations converge, since the metric of (2.4) (i) is upper bounded by its exact solution and (ii) increases monotonically across the iterations. Henceforth, for compactness in notation, the L1-BF procedure is summarized as $(\hat{\mathbf{Q}}, \hat{\mathbf{B}}) = \text{L1BF}(\mathbf{X}; \mathbf{B}(0); K)$. A pseudocode for L1-BF [35] (code available in [40]) is provided in Figure 2.1.

A state-of-the-art algorithm for L1-PCA of complex-valued data was recently presented

in [41]. L1-PCA was used for outlier identification and elimination in [42]. It was also used for robust image-fusion, face recognition, and dynamic video foreground/background extraction in [43–50]. In [51, 52] L1-PCA was used for DoA estimation. Authors in [53] proposed an L1-PCA-based nearest-subspace classifier for radar-based indoor motion recognition. L1-PCA-informed reduced-rank filtering for robust interference suppression was presented in [50]. A method for iterative re-weighted L1-PCA was most recently presented in [54]. The exact solution to L1-norm TUCKER-2 decomposition was presented in [55], and an algorithm robust decomposition of 3-way tensors based on L1-norm was proposed in [56].

2.3 Incremental and Adaptive Principal-Component Analysis

Modern big datasets often contain a very large number of measurements (data points), N, of high dimensionality (number of features), D. In such cases, batch-processing all measurements in $\mathbf{X}$ may be of prohibitively high computational cost. In some cases, the dataset to be processed is initially unavailable and data points arrive in a streaming fashion and/or the sought-after underlying signal subspace may change over time (e.g., in image/video processing [19, 57–59], dynamic face-ID [60], and DoA estimation/tracking [52, 61]). Incremental algorithm is used when the underlying signal-subspace is static and an adaptive algorithm is used when the sought after signal-subspace changes over time. Incremental processing algorithms are a subset of adaptive processing algorithms. An adaptive algorithm can also be successfully used (as an incremental algorithm) in a static subspace condition, whereas an incremental algorithm can only be used (as an adaptive algorithm) when the underlying subspace does not change. In streaming data and dynamic signal-subspace

applications, it is clear that appending every new data point to the previously collected data matrix as a new column and recalculating PCA on the augmented data matrix from scratch leads to unsustainable, continuously increasing complexity. Thus, batch PCA is rather inappropriate for processing big and/or streaming data. Similarly, the computational cost of batch L1-PCA also becomes prohibitive as N and/or D increase.

To process big and/or streaming data in an efficient way, researchers have long focused on incremental PCA solutions [62–67]. Thorough revies of incremental PCA algorithms are offered in [57, 68–70]. Similar to the batch solution, incremental PCA calculators perform well on clean or benign-noise-corrupted data (e.g., data corrupted by zero-mean, small variance additive white Gaussian noise). Conversely, incremental PCA calculators experience significant performance degradation when the processed data include any number of outliers. This observation has motivated extensive documented research on corruption-resistant incremental PCA.

Incremental algorithms inspired by the RPCA [19, 22, 29] problem formulation were proposed in [23, 70–80]. The work in [81] is an online version of robust subspace learning (RSL) in [28]. Online RPCA (OR-PCA) in [73] operates on data arriving sequentially by either accepting or rejecting new data points based on a stochastic model. Grassmannian robust adaptive subspace tracking algorithm (GRASTA) [82] operates on randomly under-sampled data matrices leading to computational improvements, while accurately tracking the underlying subspace and staying robust against sparse corruptions. The works in [47–49] offer the first incremental L1-PCA algorithms in literature, tailored to perform compressed-sensed domain video surveillance and visual tracking in videos.

This thesis work presents a complete algorithmic framework for both incremental and adaptive L1-PCA. The first algorithm computes L1-PCA incrementally, processing one measurement at a time, with low computational and memory requirements. The second algorithm revises the computed L1-PCA as new measurements arrive, demonstrating both robustness against outliers and adaptivity to signal-subspace changes and thus is appropriate for subspace-tracking applications. The proposed algorithms are evaluated in an array of experimental studies on subspace estimation, video surveillance (foreground/background separation), image conditioning, and direction-of-arrival (DoA) estimation. The sequel provides a comprehensive explanation along with pseudocodes for the proposed incremental and adaptive L1-PCA algorithms.

Chapter 3

Proposed Algorithms

3.1 Proposed Algorithm for Incremental L1-PCA (L1-IPCA)

The proposed algorithm calculates incrementally the K L1-PCs of data matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N] \in \mathbb{R}^{D \times N}$ as its columns arrive in a streaming fashion. In the case that all columns of $\mathbf{X}$ are available beforehand, the proposed algorithm processes them one-by-one for complexity savings.

To initialize, we first collect a small batch of n data points from $\mathbf{X}$, say $\mathbf{Y}_0 = [\mathbf{X}]_{:,1:n} \in \mathbb{R}^{D \times n}$ with $\text{rank}(\mathbf{Y}_0) \geq K$. Then, we run L1-BF iterations on $\mathbf{Y}_0$, with some initialization $\mathbf{B} \in \{\pm 1\}^{n \times K}$, to obtain the first approximate L1-PCA solution $(\hat{\mathbf{Q}}_0, \hat{\mathbf{B}}_0) = \text{L1BF}(\mathbf{Y}_0; \mathbf{B}; K)$.

When a new data point $\mathbf{x}_i^{(\text{in})} = [\mathbf{X}]_{:,n+i}$ arrives, $i = 1, 2, \ldots, N-n$, we first pass it through an L1-PC-informed reliability check. Specifically, similarly to [43], the *L1-reliability* of $\mathbf{x}_i^{(\text{in})}$ is defined as its angular proximity to the previously calculated L1-PCs $\hat{\mathbf{Q}}_{i-1}$,

$$r\left(\mathbf{x}_i^{(\text{in})};\ \hat{\mathbf{Q}}_{i-1}\right) = \frac{\left\|\hat{\mathbf{Q}}_{i-1}^T \mathbf{x}_i^{(\text{in})}\right\|_2^2}{\left\|\mathbf{x}_i^{(\text{in})}\right\|_2^2}. \tag{3.1}$$

Based on the outlier resistance of L1-PCA, (3.1) constitutes a measure for determining weather $\mathbf{x}_i^{(\text{in})}$ is clean (i.e., close to the nominal data subspace), or outlying/corrupted.

If $r(\mathbf{x}_i^{(\text{in})};\ \hat{\mathbf{Q}}_{i-1}) \leq \tau$, for some predetermined reliability threshold $\tau \in [0, 1)$ (practically

set close to 1), then $\mathbf{x}_i^{(\text{in})}$ is disregarded as a possible outlier and we maintain the previous L1-PCA solution $(\hat{\mathbf{Q}}_i, \hat{\mathbf{B}}_i) = (\hat{\mathbf{Q}}_{i-1}, \hat{\mathbf{B}}_{i-1})$ and the previous memory batch $\mathbf{Y}_i = \mathbf{Y}_{i-1}$. If, on the other hand, $r(\mathbf{x}_i^{(\text{in})};\ \hat{\mathbf{Q}}_{i-1}) > \tau$, then $\mathbf{x}_i^{(\text{in})}$ passes the reliability check and it is admitted for processing; the i-th L1-PCA update $(\hat{\mathbf{Q}}_i, \hat{\mathbf{B}}_i)$ is computed as follows. First, $\mathbf{x}_i^{(\text{in})}$ is appended to $\mathbf{Y}_{i-1}$, forming the augmented memory batch

$$\tilde{\mathbf{Y}}_{i-1} = \left[\mathbf{Y}_{i-1},\ \mathbf{x}_i^{(\text{in})}\right] \in \mathbb{R}^{D \times (n+1)}. \tag{3.2}$$

Then, motivated by the optimality condition in (2.6), we compute

$$\tilde{\mathbf{B}}_{i-1} = \text{sgn}\left(\tilde{\mathbf{Y}}_{i-1}^T \hat{\mathbf{Q}}_{i-1}\right) \in \{\pm 1\}^{(n+1) \times K}, \tag{3.3}$$

and use it as initialization for L1-BF iterations on $\tilde{\mathbf{Y}}_{i-1}$. At the end of the L1-BF iterations, we obtain

$$\left(\hat{\mathbf{B}}_i,\ \hat{\mathbf{Q}}_i\right) = \text{L1BF}\left(\tilde{\mathbf{Y}}_{i-1}; \hat{\mathbf{B}}_{i-1}; K\right). \tag{3.4}$$

We notice that the number of data points in memory increased from n in $\mathbf{Y}_{i-1}$ to $n+1$ in $\tilde{\mathbf{Y}}_i$. In order to maintain limited storage and computational cost, we proceed with discarding one of the points in $\tilde{\mathbf{Y}}_{i-1}$. Specifically, similar to [48], we discard the point with the minimum L1-reliability (i.e., the least angular proximity to the newly updated L1-PC subspace) as defined in (3.1). Formally, L1-IPCA identifies

$$j_{\min} = \underset{j=1,2,\ldots,n+1}{\text{argmin}}\ r\left(\left[\tilde{\mathbf{Y}}_{i-1}\right]_{:,j}; \hat{\mathbf{Q}}_i\right), \tag{3.5}$$

Proposed Algorithm L1-IPCA

Input: $\mathbf{Y}$, K, τ

$\mathbf{B} \leftarrow$ arbitrary
$(\mathbf{B}, \mathbf{Q}) \leftarrow \texttt{L1BF}(\mathbf{Y}, \mathbf{B}, K)$
When $\mathbf{x}$ arrives,
  $(\mathbf{B}, \mathbf{Q}, \mathbf{Y}) \leftarrow \texttt{uL1BF}(\mathbf{Y}, \mathbf{x}, \mathbf{Q}, K, \tau)$

Output: $(\hat{\mathbf{B}}, \hat{\mathbf{Q}}) \leftarrow (\mathbf{B}, \mathbf{Q})$

Function: $(\mathbf{B}, \mathbf{Q}, \mathbf{Y}, \tau) \leftarrow \texttt{uL1BF}(\mathbf{Y}, \mathbf{x}, \mathbf{Q}, K, \tau)$

$r \leftarrow \texttt{L1rel}(\mathbf{x}, \mathbf{Q})$
if $r > \tau$
  $\tilde{\mathbf{Y}} \leftarrow [\mathbf{Y}, \mathbf{x}]$
  $\mathbf{B} \leftarrow \text{sgn}(\tilde{\mathbf{Y}}^T \mathbf{Q})$
  $(\mathbf{B}, \mathbf{Q}) \leftarrow \texttt{L1BF}\ (\tilde{\mathbf{Y}}, \mathbf{B}, K)$
  $r_j \leftarrow \texttt{L1rel}([\tilde{\mathbf{Y}}]_{:,j}, \mathbf{Q})$, $j = 1, \ldots, n+1$
  $j_{\min} \leftarrow \text{argmin}_{j=1,\ldots,n+1}\, r_j$
  $\mathbf{Y} \leftarrow [\tilde{\mathbf{Y}}]_{:,\{1,\ldots,n+1\} \setminus j_{\min}}$
Return $\mathbf{B}$, $\mathbf{Q}$, $\mathbf{Y}$

Function: $r \leftarrow \texttt{L1rel}\ (\mathbf{x}, \mathbf{Q})$

$r \leftarrow \frac{||\mathbf{Q}^T \mathbf{x}||_2^2}{||\mathbf{x}||_2^2}$
Return r

Figure 3.1: Pseudocode of proposed L1-IPCA algorithm.

and discards the $(j_{\min})$-th column of $\tilde{\mathbf{Y}}_{i-1}$, setting the i-th memory matrix

$$\mathbf{Y}_i = \left[\tilde{\mathbf{Y}}_{i-1}\right]_{:,\{1,\ldots,n+1\} \setminus j_{\min}} \in \mathbb{R}^{D \times n}. \tag{3.6}$$

In view of the above, the proposed algorithm has multiple lines of defense against outliers. First, L1-IPCA starts with calculating the L1-PCs of a small original batch. These L1-PCs, being robust against any outliers in the original batch, set a first measure of reliability for future processed points. Then, the reliability of an incoming point is evaluated by means of the previously computed L1-PCs, thus protecting the incremental L1-PCA procedure against processing outliers. Finally, any point that passes the reliability check is processed by the robust L1-BF procedure. A detailed description of the proposed L1-IPCA is provided in Figure 3.1.

Complexity. According to [35], L1-BF returns the K (approximate) L1-PCs of $\tilde{\mathbf{Y}}_i \in \mathbb{R}^{D \times n}$ with cost $O(nD \min\{n, D\} + n^2K^2(K^2 + \min\{n, D\}))$, for any i. Since i takes values $1, 2, \ldots, N - n$, the total cost of L1-IPCA is $O(NnD \min\{n, D\} + Nn^2K^2(K^2 + \min\{n, D\}))$, linear in N. That is, if $n > D$, then the cost is $O(Nn^2K^2(K^2 + D))$; on the other hand, if $D \geq n$, the cost is $O(Nn^2(K^4 + K^2n + D))$.

Comparison with [48]. At this point, it is worth noting that the pioneering work in [48] also proposed L1-BF updates for incremental L1-PCA in compressed-sensed-domain video surveillance. The proposed L1-IPCA algorithm differs from the one in [48] in three main ways. First, instead of (3.3), the algorithm of [48] sets the L1-BF-initialization matrix to

$$\tilde{\mathbf{B}}_i = \left[\hat{\mathbf{B}}_{i-1}^T, \mathbf{b}_{\text{exact}}\right]^T, \tag{3.7}$$

where

$$\mathbf{b}_{\text{exact}} = \underset{\mathbf{b} \in \{\pm 1\}^K}{\text{argmax}} \left\| \tilde{\mathbf{Y}}_{i-1} \left[\hat{\mathbf{B}}_{i-1}^T, \mathbf{b}_{\text{exact}}\right]^T \right\|_*. \tag{3.8}$$

To identify $\mathbf{b}_{\text{exact}}$, [48] first finds $\tilde{\mathbf{Y}}_{i-1}\hat{\mathbf{B}}_{i-1}^T$ with cost $O(KnD)$; then, for each of the 2^K candidate solutions in $\{\pm 1\}^K$, it performs an SVD of a $D \times K$ matrix. Thus, initializing L1-BF as in [48] costs $O(KnD + 2^K DK^2)$. The proposed initialization in (3.3) attains similar performance and costs only $O(KnD)$. Secondly, the memory-batch size-preserving step removes the entry of the memory batch that lies farthest from the current L1-PCs by identifying

$$j_{min} = \underset{1 \leq j \leq n}{\text{argmin}} ||\mathbf{y}_j - \hat{\mathbf{Q}}\hat{\mathbf{Q}}^T\mathbf{y}_j||_F^2, \tag{3.9}$$

where $\mathbf{y}_j = [\mathbf{Y}_{i-1}]_{:,j}$ and discards the $(j_{\min})$-th column of $\tilde{\mathbf{Y}}_{i-1}$ to obtain $\tilde{\mathbf{Y}}_i$, by setting the

i-th memory matrix $\mathbf{Y}_i = \left[\tilde{\mathbf{Y}}_{i-1}\right]_{:,\{1,\ldots,n\}\setminus j_{\min}} \in \mathbb{R}^{D\times n}$ before PC-update. However, L1-IPCA discards the entry of $\tilde{\mathbf{Y}}_{i-1}$ with the least L1-reliability value to obtain $\tilde{\mathbf{Y}}_i$ after PC-update. A third important difference is that the proposed algorithm updates the L1-PCA solution only on incoming points that pass the L1-PC-informed reliability check ([48] processes every incoming point). Thus, L1-IPCA has an additional line of defense against outliers in the processed data.

3.2 Proposed Algorithm for Adaptive L1-PCA (L1-APCA)

L1-IPCA presented above is tailored to cases that the sought-after signal subspace is constant across all processed data. In many signal processing applications however it is desired to track a dynamic signal subspace that changes across the collected data points (e.g., in direction-of-arrival tracking). To this end, an algorithm for adaptive L1-PCA (L1-APCA), derived by two main modifications of L1-IPCA is proposed as follows.

3.2.1 Modification 1: Reliability Threshold Adjustment

When the signal subspace changes significantly, new incoming points may fail the reliability check and be inserted to the secondary memory or discarded. Assuming that outliers appear rather sporadically, when multiple incoming points fail the reliability check one after the other, then this is a strong indication that the signal subspace has changed. Thus, in L1-APCA we consider adjustable reliability threshold that decreases every time an incoming point fails the reliability check and resets whenever an incoming point is admitted for processing.

Specifically, let threshold τ_i denote the reliability threshold by which the i-th incoming point $\mathbf{x}_i^{(\text{in})}$ is evaluated, with initialization $\tau_1 = \tau$. If $r\left(\mathbf{x}_i^{(\text{in})};\ \hat{\mathbf{Q}}_{i-1}\right) < \tau_i$ and $\mathbf{x}_i^{(\text{in})}$ fails the reliability check, then we reduce $\tau_{i+1} = \tau_i \rho$, for some predetermined decrease ratio ρ in $(0, 1]$. Clearly, $\rho = 1$ corresponds to fixed threshold, as used in L1-IPCA. If $\mathbf{x}_i^{(\text{in})}$ passes the reliability check, then τ_{i+1} is reset to τ.

3.2.2 Modification 2: Preserve Recent Measurements

Consider a significant change of the nominal signal subspace. Consider also that an incoming point $\mathbf{x}_i^{(\text{in})}$ from the new signal subspace passes the reliability check (possibly after threshold reduction) and is inserted in $\tilde{\mathbf{Y}}_{i-1}$. $\mathbf{x}_i^{(\text{in})}$ will contribute to an update of the PCs from $\hat{\mathbf{Q}}_{i-1}$ to $\hat{\mathbf{Q}}_i$. However, if most of the points in $\tilde{\mathbf{Y}}_{i-1}$ are drawn from the previous/old signal subspace, then the new PCs in $\hat{\mathbf{Q}}_i$ may remain almost invariant. Thus, when the reliability of points in memory $\tilde{\mathbf{Y}}_{i-1}$ is evaluated by means of $\hat{\mathbf{Q}}_i$ as in (3.5), $\mathbf{x}_i^{(\text{in})}$ may be found to be the least coherent point and as such be discarded from $\tilde{\mathbf{Y}}_{i-1}$. Most certainly, such an event would inhibit the subspace tracking process. Therefore, in L1-APCA, we revise steps in (3.5)-(3.6) so that the $q < n$ most recently added measurements are not dropped from $\tilde{\mathbf{Y}}_{i-1}$. That is, the i-th memory matrix is defined as $\mathbf{Y}_i = \left[\tilde{\mathbf{Y}}_{i-1}\right]_{:,\{1,\ldots,n+1\}\setminus j_{\min}} \in \mathbb{R}^{D\times n}$ where

$$j_{\min} = \underset{j=1,2,\ldots,n+1-q}{\operatorname{argmin}} \; r\left(\left[\tilde{\mathbf{Y}}_{i-1}\right]_{:,j}; \hat{\mathbf{Q}}_i\right). \tag{3.10}$$

A detailed description of L1-APCA, including all above modifications, is provided in the pseudocode of Figure 3.2.

Proposed Algorithm L1-APCA

Input: $\mathbf{Y}$, K, q, $\tau_{\max}$, ρ

$\mathbf{B} \leftarrow$ arbitrary, $\tau \leftarrow \tau_{\max}$
$(\mathbf{B}, \mathbf{Q}) \leftarrow \texttt{L1BF}(\mathbf{Y}, \mathbf{B}, K)$
When $\mathbf{x}$ arrives,
  $(\mathbf{B}, \mathbf{Q}, \mathbf{Y}, \tau) \leftarrow \texttt{aL1BF}(\mathbf{Y}, \mathbf{x}, \mathbf{Q}, K, q, \tau_{\max}, \tau, \rho)$

Output: $(\hat{\mathbf{B}}, \hat{\mathbf{Q}}) \leftarrow (\mathbf{B}, \mathbf{Q})$

Function: $(\mathbf{B}, \mathbf{Q}, \mathbf{Y}, \mathbf{S}, \tau) \leftarrow \texttt{aL1BF}(\mathbf{Y}, \mathbf{x}, \mathbf{S}, \mathbf{Q}, K, m, q, \tau_{\max}, \tau, \rho)$

$r \leftarrow \texttt{L1rel}(\mathbf{x}, \mathbf{Q})$
if $r > \tau$
  $\tilde{\mathbf{Y}} \leftarrow [\mathbf{Y}, \mathbf{x}]$, $\tau \leftarrow \tau_{\max}$
  $\mathbf{B} \leftarrow \text{sgn}(\tilde{\mathbf{Y}}^T \mathbf{Q})$
  $(\mathbf{B}, \mathbf{Q}) \leftarrow \texttt{L1BF}\ (\tilde{\mathbf{Y}}, \mathbf{B}, K)$
  $r_j \leftarrow \texttt{L1rel}([\tilde{\mathbf{Y}}]_{:,j}, \mathbf{Q})$, $j = 1, \ldots, n+1$
  $j_{\min} \leftarrow \text{argmin}_{j=1,\ldots,n+1-q}\, r_j$
  $\mathbf{Y} \leftarrow [\tilde{\mathbf{Y}}]_{:,\{1,\ldots,n+1\}\setminus j_{\min}}$
else
  $\tau \leftarrow \tau\rho$
Return $\mathbf{B}$, $\mathbf{Q}$, $\mathbf{Y}$, τ

Function: $r \leftarrow \texttt{L1rel}\ (\mathbf{x}, \mathbf{Q})$

$r \leftarrow \frac{||\mathbf{Q}^T \mathbf{x}||_2^2}{||\mathbf{x}||_2^2}$
Return r

Figure 3.2: Pseudocode of proposed L1-APCA algorithm.

Chapter 4

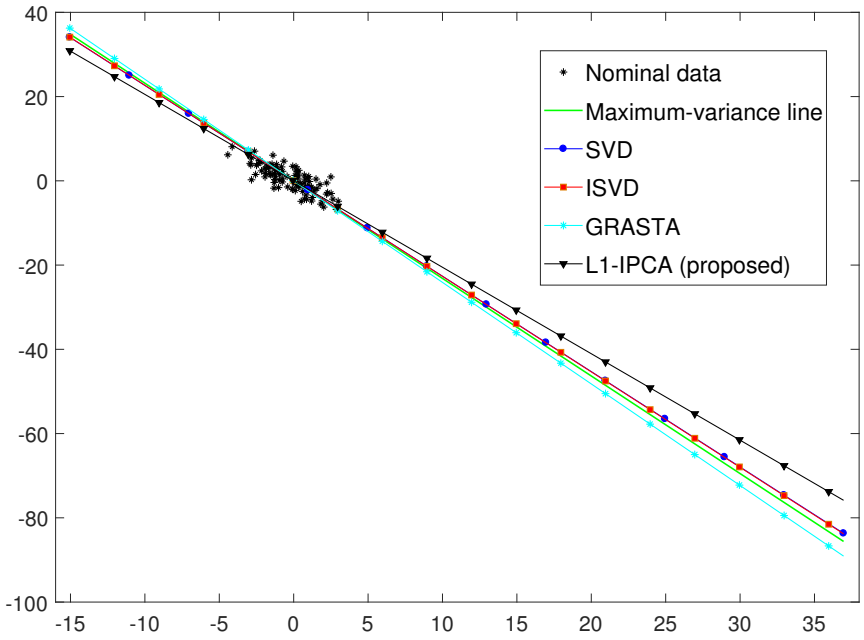
Numerical and Experimental Studies

4.1 Synthetic Data Analysis

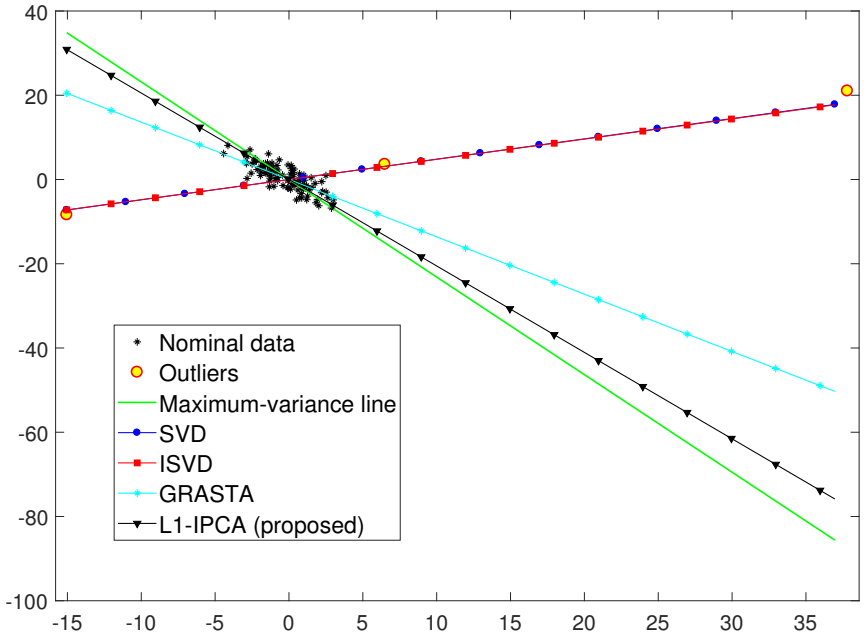
4.1.1 Toy Example – Line-Fitting

The performance of the proposed L1-IPCA algorithm is first evaluated with a line-fitting experiment. Consider $\mathbf{z} \in \mathbb{R}^{(D=2)\times(K=1)}$, with $\|\mathbf{z}\|_2 = 1$ and $\alpha = 10$. $N = 100$ data points are drawn from $\mathcal{N}(\mathbf{0}_2, \alpha\mathbf{z}\mathbf{z}^T)$ to form matrix $\mathbf{X} \in \mathbb{R}^{2\times 100}$. Every entry of $\mathbf{X}$ is corrupted with zero-mean additive white Gaussian noise (AWGN) of variance 1 from $\mathcal{N}(0, 1)$; this results in rank$(\mathbf{X}) = D = 2$. To approximate $\mathbf{z}$, L1-IPCA is run on $\mathbf{X}$ for $K = 1$, setting memory batch-size $n = 10$ and L1-reliability threshold $\tau = 0.85$. In Figure 4.1(a), the nominal data points (black asterisks) and the L1-PC obtained by L1-IPCA after processing all N points are plotted. In the same figure, the lines obtained by SVD, incremental singular-value-decomposition (ISVD) [62], and GRASTA [82] are plotted. It is observed that all methods perform similarly, approximating well the line defined by $\mathbf{z}$ (maximum-variance line).

Then, columns 5, 57 and 74 of $\mathbf{X}$ are corrupted by adding to them corruption from $\mathcal{N}(\mathbf{0}_2, \beta\mathbf{p}\mathbf{p}^T)$, where $\beta = 40\alpha$ and $\mathbf{p} \in \mathbb{R}^{2\times 1}$ is such that $||\mathbf{p}||_2 = 1$ and $\arccos(\mathbf{p}^T\mathbf{z}) = 78°$. L1-IPCA is run again on $\mathbf{X}$ for $K = 1$, keeping $n = 10$ and $\tau = 0.85$. In Figure 4.1(b), the L1-PC obtained by L1-IPCA after processing all N data points is plotted. In addition, the



(a)



(b)

Figure 4.1: Line-fitting experiment. PC calculation on (a) clean/nominal and (b) outlier-corrupted data; $N = 100$, $D = 2$, $K = 1$, $n = 10$, and $\tau = 0.85$.

lines obtained by SVD, ISVD [62], and GRASTA [82] are also plotted. This time, the L2-norm based methods (SVD, ISVD) are significantly misled by the corrupted/outlying data. GRASTA [82] displays some deviation. Interestingly, the proposed L1-IPCA algorithm remains almost unaffected by the outliers and very close to the maximum-variance line of $\mathbf{z}$.

4.1.2 Incremental subspapce estimation with L1-IPCA

Next, the performance of L1-IPCA in intermediate incremental updates is evaluated. Specifically, the normalized subspace error of $\hat{\mathbf{q}}_i$ (i.e., the approximate L1-PC after $\mathbf{x}_i^{(\text{in})}$ is processed) is measured as

$$e_i = \frac{1}{2}||\mathbf{z}\mathbf{z}^T - \hat{\mathbf{q}}_i\hat{\mathbf{q}}_i^T||^2 \in [0, 1]. \tag{4.1}$$

In this experiment, $D = 5$ and $N = 200$ points are drawn from the nominal distribution $\mathcal{N}(\mathbf{0}_5, \alpha\mathbf{z}\mathbf{z}^T)$, for $\alpha = 55$, $||\mathbf{z}||_2 = 1$, to form data matrix $\mathbf{X} \in \mathbb{R}^{5\times 200}$. All entries of $\mathbf{X}$ are also corrupted by AWGN from $\mathcal{N}(0, 1)$. Columns 7, 60, 125 and 170 of $\mathbf{X}$ are also corrupted additively by outliers from $\mathcal{N}(\mathbf{0}_5, \beta\mathbf{p}\mathbf{p}^T)$ where $\beta = 30\alpha$, $\mathbf{p} \in \mathbb{R}^{5\times 1}$, $||\mathbf{p}||_2 = 1$, and $\arccos(\mathbf{p}^T\mathbf{z}) = 74.33°$. Setting $n = 20$, $K = 1$, and $\tau = 0.66$, the proposed L1-IPCA algorithm is run while evaluating e_i for every i. The average value of $\{e_i\}_{i=1}^{N-n}$ over 2000 independent realizations of nominal data points, noise, and outlier corruption is calculated. In Figure 4.2, e_i vs. update index i is plotted. Together with the proposed algorithm, the performance of standard PCA (SVD), batch-calculated jointly on $[\mathbf{X}]_{:,1:i+n}$ (i.e., for index i, we carry out SVD on entire $[\mathbf{X}]_{:,1:i+n}$), L1-BF [50], also batch-calculated on $[\mathbf{X}]_{:,1:i+n}$, ISVD [62], GRASTA [82,83], OR-PCA [73,74], the method of [49] and the method of [47]

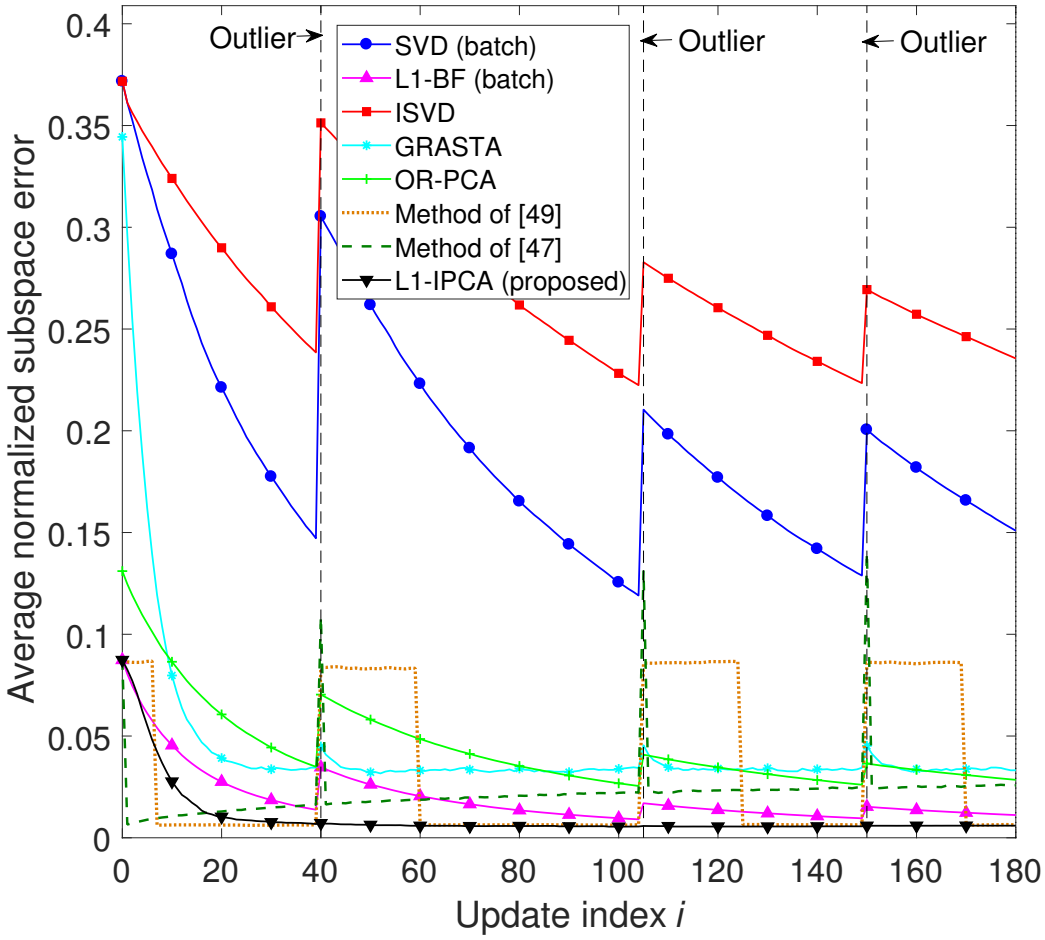


Figure 4.2: Subspace estimation experiment. Average normalized subspace error versus update index i; $D = 5$, $N = 200$, $K = 1$; $n = 20$, $\tau = 0.66$.

are also plotted[1].

Clearly ISVD [62] and SVD start at a relatively high normalized error due to the presence of one outlier-corrupted point in $[\mathbf{X}]_{:,1:(n=20)}$ (point $[\mathbf{X}_{:,7}]$). These methods exhibit improvement as they process nominal points. However, when they encounter another outlier $[\mathbf{X}_{:,60}]$, $[\mathbf{X}_{:,125}]$ and $[\mathbf{X}_{:,170}]$ (dashed vertical line), they deviate again from the nominal

[1]Open access MATLAB codes were found in the GitHub library [84]; the MATLAB implementation of ReProCS [72] was available in [85]; and the MATLAB implementation of GRASTA [82] is available in [86].

subspace. GRASTA subspace error starts at 0.3 and drops to 0.035, quickly reaching lower error and remains at this level throughout the remaining updates, almost unaffected by the outliers. L1-BF, the method of [49] and the method of [47] start at an error of about 0.08 and OR-PCA starts at an error close to 0.12. The method of [47] quickly drops to a lower error because it finds the exact bit-flipping vector, $\mathbf{b}_{\text{exact}}$, when a new point arrives, through exhaustive search. L1-BF and OR-PCA drop towards lower error but show some responsiveness to outliers. The method of [49] and the method of [47] also respond to outliers. The method of [49] drops to low error after processing $n = 20$ nominal points since it encountered an outlier and the method of [47] drops to low error very quickly after processing 1 nominal point since it encountered an outlier. It is observed that the average error of the method of [47] monotonically increases throughout all updates, owing to the memory-batch size-preserving step (3.9). The proposed L1-IPCA algorithm starts from an error of 0.08 due its initialization to the L1-PC of $[\mathbf{X}]_{:,1:(n=20)}$ (obtained by L1-BF). During the incremental updates, the proposed algorithm converges fast to very low subspace error (close to 0) and remains there throughout all the updates, staying practically unaffected by the outlier-corrupted points in $\mathbf{X}$, thus outperforming every counterpart throughout all updates. With its L1-reliability check feature, L1-IPCA strives to avoid processing outliers. For the same experiment described above, the frequency with which $[\mathbf{X}]_{:,i}, i = n+1, \ldots, N$, passes the reliability check is plotted in Figure 4.3. Noticeably, for $\tau = 0.66$ all nominal points pass the L1-reliability check more that 70% of the time and is admitted for processing. On the other hand, L1-IPCA manages to detect and discard the outliers $[\mathbf{X}_{:,60}]$, $[\mathbf{X}_{:,125}]$ and $[\mathbf{X}_{:,170}]$ more than 90% of the time.

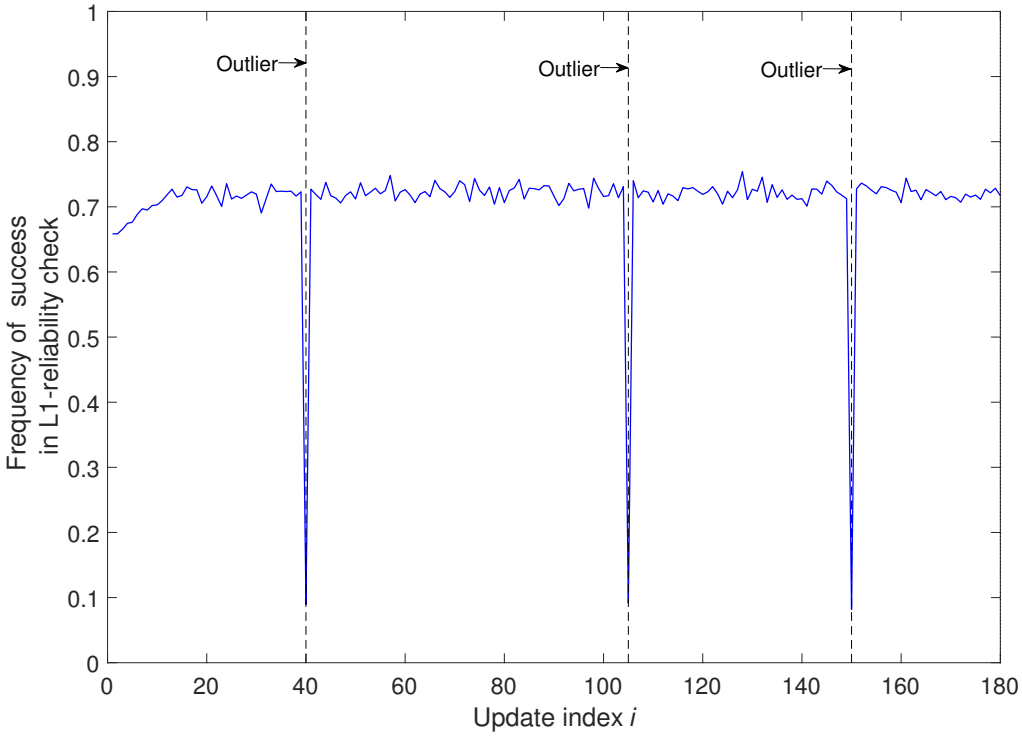


Figure 4.3: Subspace estimation experiment. Frequency of success versus update index i.

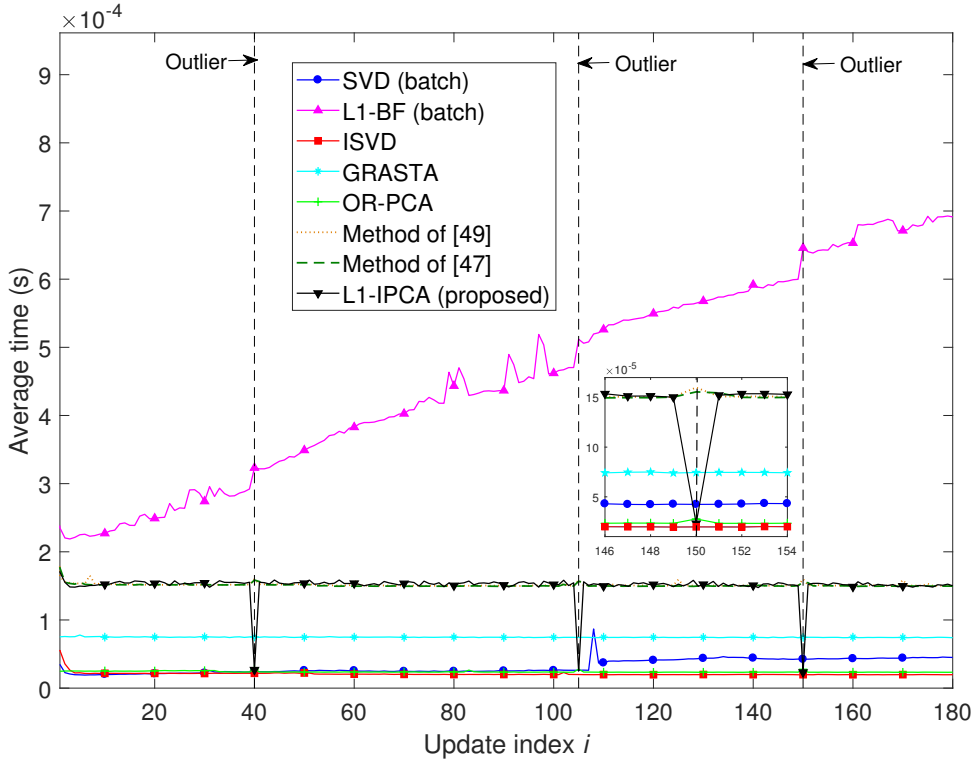


Figure 4.4: Subspace estimation experiment. Average time versus update index i.

Next, for the same study and same algorithms, the average computation time a each update step is plotted in Figure 4.4 [2]. Clearly L1-BF (batch processing) has a higher computational cost, expectedly increasing across i; interestingly, when L1-BF processes outliers, its computation effort increases as more bit-flipping iterations are needed for convergence. Batch SVD has lower cost, also monotonically increasing with i. All incremental methods need very low average computation time below 155μs, for every i. Interestingly, the cost of L1-IPCA drops to 10μs for $i = 60 - n$, $i = 125 - n$ and $i = 170 - n$, since 90% of the time $[\mathbf{X}]_{:,60}$, $[\mathbf{X}]_{:,125}$ and $[\mathbf{X}]_{:,170}$ are not admitted for processing.

[2]Reported computation times are measured in MATLAB R2017a, run on a computer equipped with Intel(R) core(TM) i7-6700 processor 3.40GHz and 32GB RAM.

4.1.3 Subspace tracking with L1-APCA

Next, we evaluate the performance of the proposed adaptive L1-PCA algorithm (L1-APCA) for subspace tracking. We consider data matrix $\mathbf{X} \in \mathbb{R}^{5\times 250}$, the first 130 columns of which are drawn from $\mathcal{N}(\mathbf{0}_5, \alpha\mathbf{z}\mathbf{z}^T)$, where $||\mathbf{z}||_2 = 1$ and α is the same as in the study of Figure 4.2. The last 120 columns are drawn from $\mathcal{N}(\mathbf{0}_5, \alpha\mathbf{z}'\mathbf{z}'^T)$, where $\mathbf{z}' \in \mathbb{R}^{5\times 1}$, $||\mathbf{z}'||_2 = 1$ and $\arccos(\mathbf{z}'^T\mathbf{z}) = 32.43°$. That is, the nominal subspace changes after the 130 first points are processed. All entries of $\mathbf{X}$ are corrupted by AWGN from $\mathcal{N}(0, 1)$. Columns 7, 60 and 210 of $\mathbf{X}$ are once again corrupted additively by outliers from $\mathcal{N}(\mathbf{0}_5, \beta\mathbf{p}\mathbf{p}^T)$, where $||\mathbf{p}||_2 = 1$ and β is the same as in the previous experiment. $\arccos(\mathbf{p}^T\mathbf{z}) = 84.26°$ and $\arccos(\mathbf{p}^T\mathbf{z}') = 77.53°$. We set L1-APCA parameters $n = 20$, $\tau = 0.8$, threshold decrease ratio $\rho = 0.5$, and number of maintained recent points $q = 0.75n$ (i.e., at each adaptation instance, we preserve in the memory matrix 75% of the points processed most recently). In Figure 4.5 we plot the normalized subspace error calculated as $e_i = \frac{1}{2}||\mathbf{z}\mathbf{z}^T - \hat{\mathbf{q}}_i\hat{\mathbf{q}}_i^T||^2$ for $i \leq 130$ and $e_i = \frac{1}{2}||\mathbf{z}'\mathbf{z}'^T - \hat{\mathbf{q}}_i\hat{\mathbf{q}}_i^T||^2$ for $i > 130$. Together with L1-APCA, we plot the performance of SVD (batch), L1-BF (batch), ISVD, GRASTA, and OR-PCA. Noticeably the L2-norm based methods deviate from the nominal subspace when they process outliers and display slower response to subspace changes. GRASTA, OR-PCA and L1-BF are relatively robust against outliers compared to the L2-norm based methods. L1-APCA outperforms all counterparts, exhibiting fast convergence to the first nominal subspace ($\mathbf{z}$), sturdy outlier resistance, and fast adaptation to the second nominal subspace $\mathbf{z}'$. GRASTA adapts slightly faster than L1-APCA to the subspace change, converging though to higher subspace error. In Figure 4.7, the frequency of success of $[\mathbf{X}]_{:,i}$, $i = n + 1, \ldots, N$ in the reliability check is

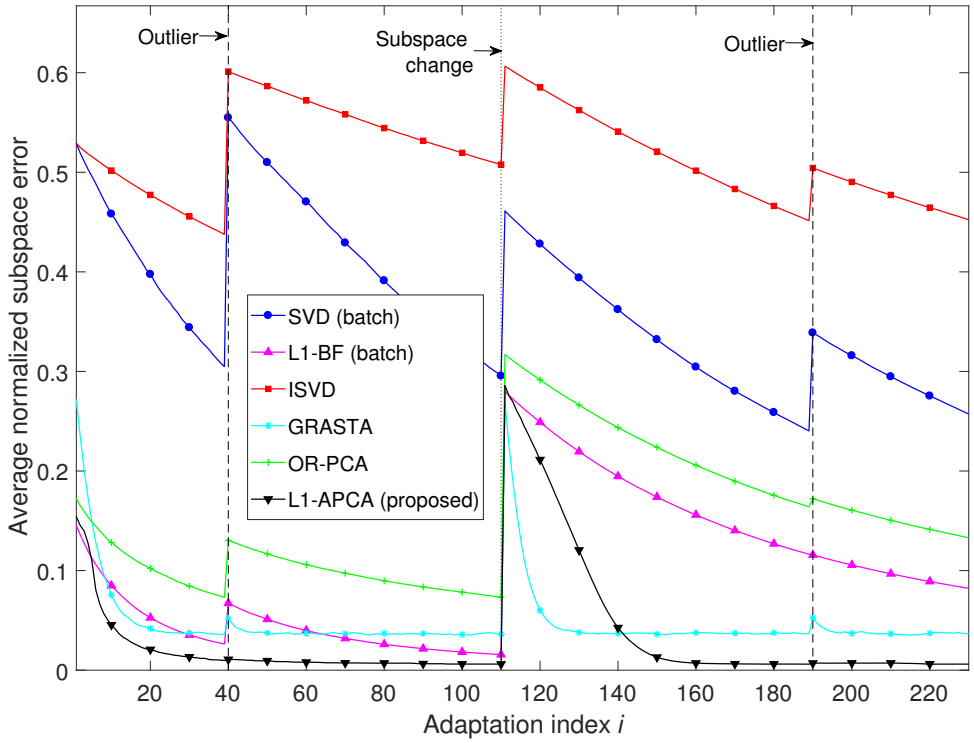


Figure 4.5: Subspace tracking experiment. Average normalized subspace error versus adaptation index i; $D = 5$, $N = 250$, $K = 1$; $n = 20$, $\tau_{max} = 0.8$, $\rho = 0.35$ and $q = 0.75n$; Subspace change after $130 - n$ points.

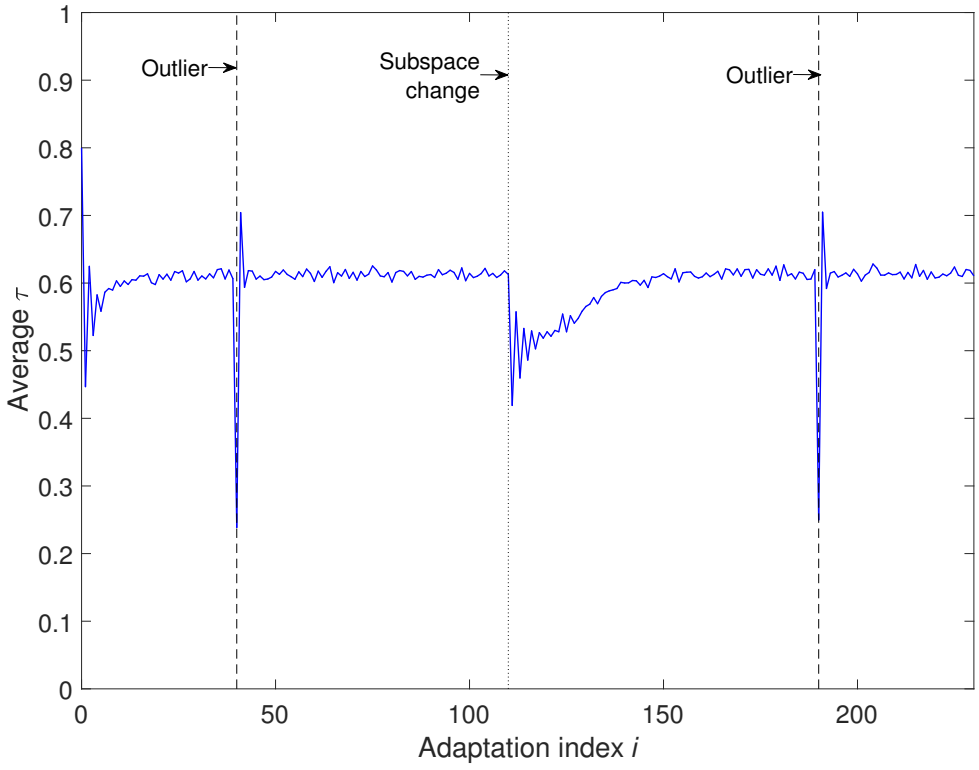


Figure 4.6: Subspace tracking experiment. L1-reliability threshold τ_i versus adaptation index i.

plotted. This plot is best interpreted together with Figure 4.6, where the average value of τ_i versus i is plotted. Clearly, for increased maximum threshold $\tau_{max} = 0.8$, most nominal points are tested with threshold close to 0.6 and exhibit frequency of success (i.e., any point being able to participate in PC-adaptation) close to 0.65. Once again, both outliers are identified and discarded more than 90% of the time. When an outlier is discarded, τ_i is decreased to 35% of τ_{max}. Expectedly, the first few points points from the new subspace are often discarded; though, dropping the threshold with reduction factor $\rho = 0.35$ allows for quick adaptation.

In Figure 4.8, the average update time for each algorithm versus i is plotted. Once again, L1-BF is the most computationally expensive algorithm and its cost increases when it processes outliers. On the other hand, all incremental methods update on average in less than 0.2ms and again, L1-reliability success rates affect the execution time of the proposed L1-APCA algorithm (similar to Figure 4.4).

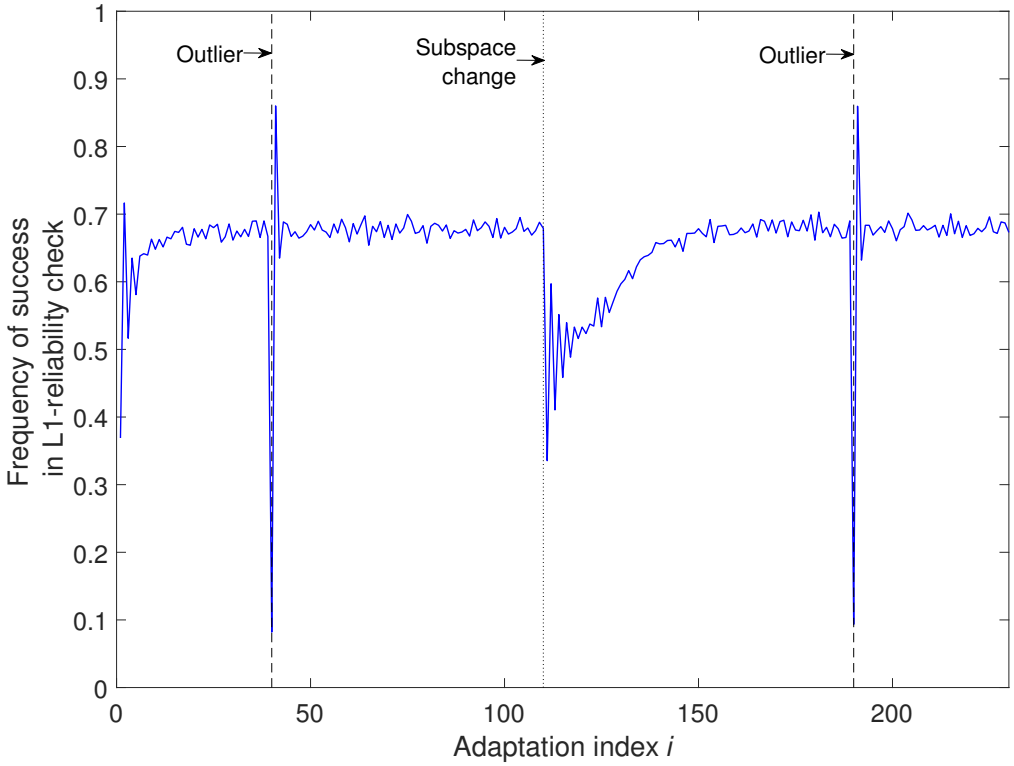


Figure 4.7: Subspace tracking experiment. Frequency of success versus adaptation index i.

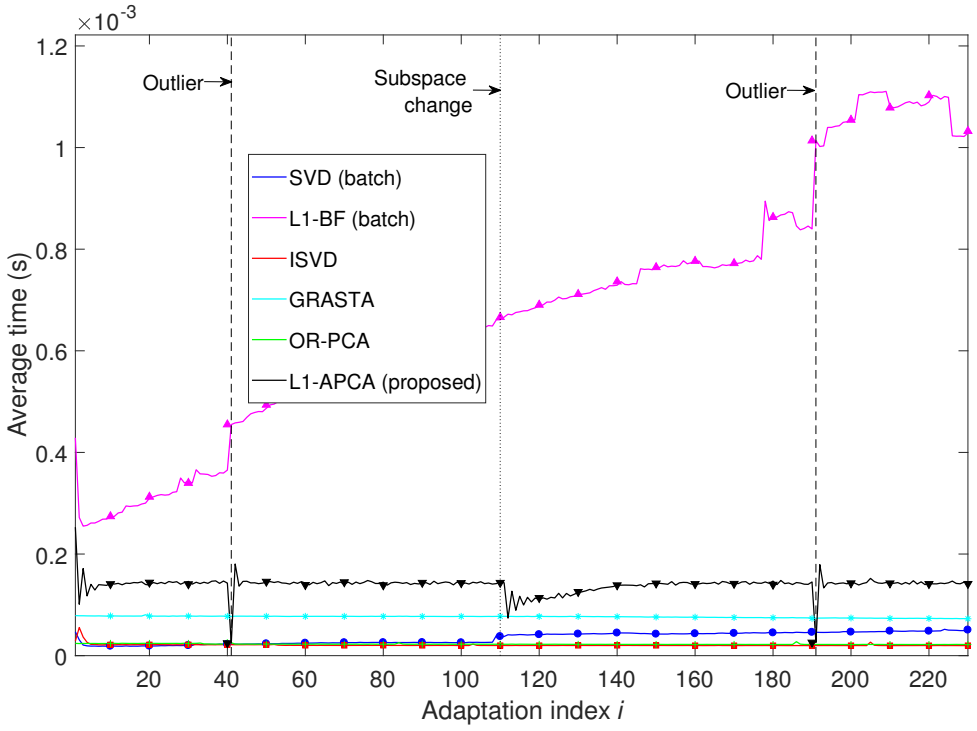


Figure 4.8: Subspace tracking experiment. Average time versus adaptation index i.

4.2 Image Conditioning

In this experiment, conditioning of face images is performed. An interesting proposition was made by Barsi and Jacobs in [87] stating that images (convex-Lambertian) taken under varying, distant illumination lie near an approximately nine-dimensional (low-dimensional) subspace known as the harmonic-plane. This proposition motivates the glare/shadow artifacts removal experiment wherein we can approximate the image accurately by a low-dimensional subspace. Specifically, we operate on images of a person's face from the PICS database [88] captured in varying illumination conditions that resulted in unwanted glare and shadow artifacts. 14 images of a single individual captured under varying illumination are chosen and cropped to 200×200 pixels each. Each image is then vectorized and stacked one next to the other as columns of data matrix $\mathbf{X} \in \mathbb{R}^{40000 \times 14}$. In this experiment, the face characteristics form the sought-after static background whereas the illumination variations (glare and shadow artifacts) constitute foreground outliers that we wish to eliminate. We set $n = 5$ and $\tau = 0.95$ and run the proposed L1-IPCA algorithm to obtain $K = 5$ approximate L1-PCs after processing all 14 images. We remove unwanted illumination artifacts from each vectorized image $\mathbf{x}_i = [\mathbf{X}]_{:,i}$ by projecting it on the span of calculated L1-PCs, $\hat{\mathbf{Q}}$ as $\hat{\mathbf{Q}}\hat{\mathbf{Q}}^T\mathbf{x}_i$. In Figure 4.9 we present (a) an original face instance with glare and shadows, and the same image conditioned by (b) ISVD [62], (c) the method of [64], (d) GRASTA [82], (e) PCP [19], (f) OR-PCA [73, 74], (g) the method of [49], (h) the method of [47] and (i) L1-IPCA (proposed). We observe that ISVD [62] and the method of [64] retain most glare. GRASTA [82], PCP [19], OR-PCA [73, 74], and the method of [49] perform improved glare/shadow elimination. The proposed L1-IPCA algorithm and the

Figure 4.9: Image conditioning experiment. (a) Original face image with glare and shadows. Image conditioned with (b) ISVD [62], (c) the method of [64], (d) GRASTA [82], (e) PCP [19], (f) OR-PCA [73, 74], (g) the method of [49], (h) the method of [47] and (i) L1-IPCA (proposed).

method of [47] demonstrate superior image conditioning.

Next, the experiment is repeated on a different set of face images. 13 images of a different face under varying illumination are obtained from the same PICS database. Each image is cropped to 200×200 pixels and vectorized to form the data matrix $\mathbf{X} \in \mathbb{R}^{40000 \times 13}$. We

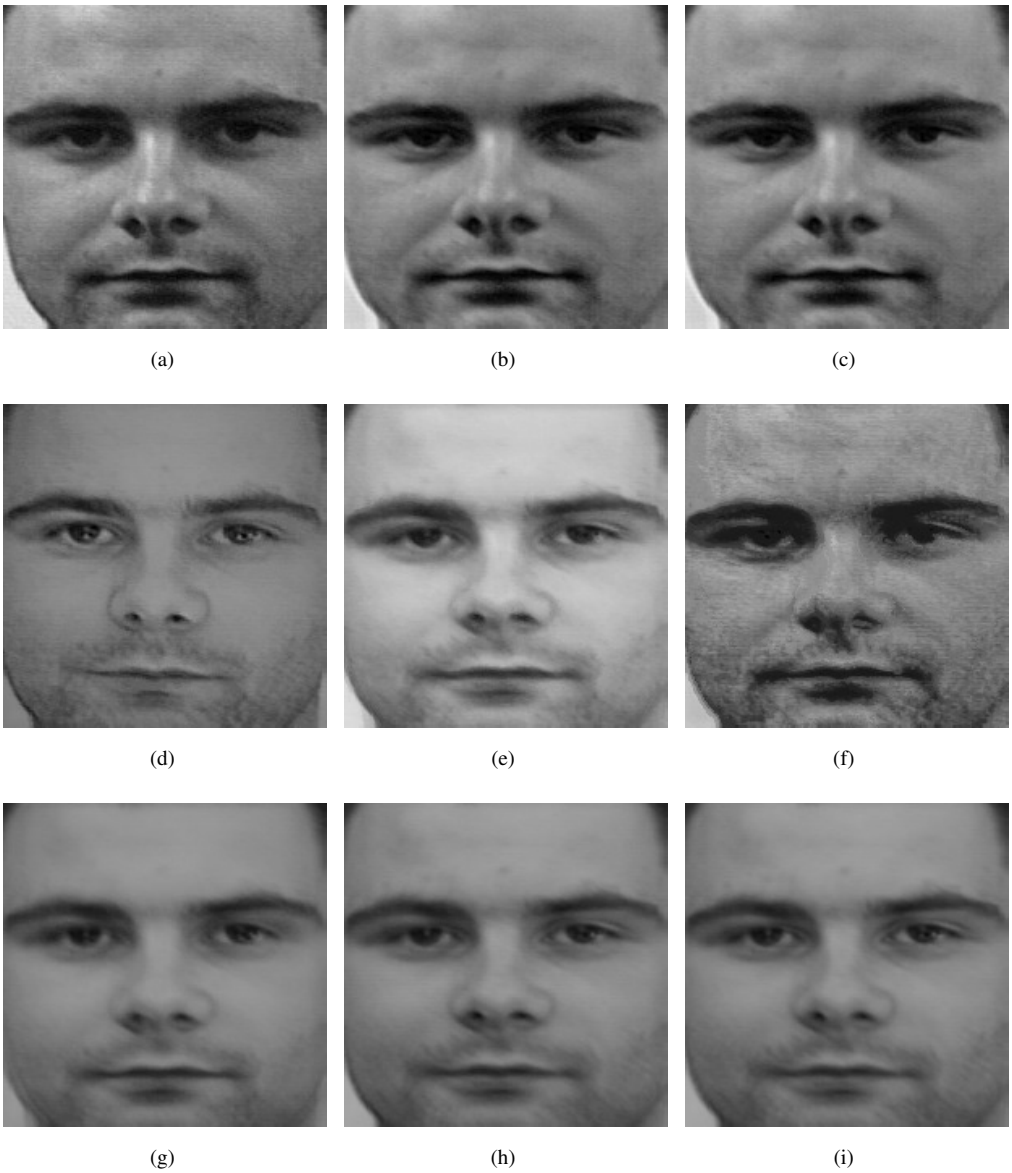


Figure 4.10: Image conditioning experiment. (a) Original face image with glare and shadows. Image conditioned with (b) ISVD [62], (c) the method of [64], (d) GRASTA [82], (e) PCP [19], (f) OR-PCA [73, 74], (g) the method of [49], (h) the method of [47] and (i) L1-IPCA (proposed).

re-run the experment by setting $n = K = 4$ and $\tau = 0.975$. In Figure 4.10 we present (a) an original face instance with glare and shadows, and the same image conditioned by (b) ISVD [62], (c) the method of [64], (d) GRASTA [82], (e) PCP [19], (f) OR-PCA [73, 74], (g) the method of [49], (h) the method of [47] and (i) L1-IPCA (proposed). We observe

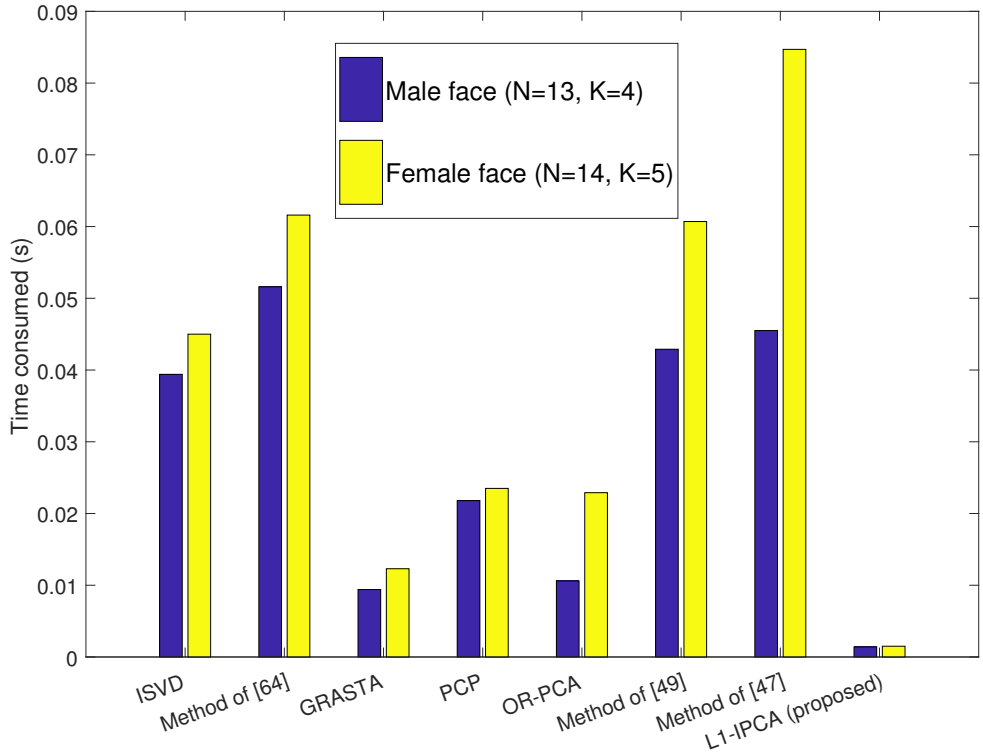


Figure 4.11: Time consumed for image conditioning

similar performance compared to the previous case, i.e, ISVD [62] and the method of [64] retain most glare. PCP [19], OR-PCA [73, 74] perform improved glare/shadow elimination. GRASTA [82], the method of [49], the method of [47] and the proposed L1-IPCA algorithm demonstrate superior image conditioning.

The time required by each method to compute the underlying subspace $\hat{\mathbf{Q}}$ (onto which each image is projected for glare removal) is computed and plotted as a bar-graph in Figure 4.11. It is observed that, for processing the male face ($K = 4$; see blue bars), method of [64], method of [47], method of [49], ISVD consume similarly higher time, followed by PCP with lower time consumed. GRASTA and OR-PCA consume even lower time whereas the proposed L1-IPCA algorithm consumes the least time. For processing the female face

($K = 5$; see yellow bars), each method consumes proportionally higher time. However the proposed L1-IPCA algorithm displays strikingly fast performance across the board. Therefore, the image conditioning experiment concludes that L1-IPCA performs superior glare/shadow artifacts removal at strikingly fast speeds.

4.3 Background/Foreground Separation in Video Sequences

Video foreground extraction is an important computer vision application used, e.g., in real-time gesture/object identification, human-computer interaction, security surveillance, traffic monitoring, and optical-motion capture [89, 90]. The background of each frame forms the static nominal subspace while moving foreground components (e.g., people and vehicles) constitute intermittent outliers. The foreground components of the video sequence are typically extracted by first estimating the underlying background of the video and then subtracting it from the original frame. For this experiment, we use a surveillance video recorded at a shopping center in Portugal, available in the standard CAVIAR database [91]. The video consists of $N = 474$ frames of size 202 by 269 pixels. Video processing is carried out as follows. We cut the video so that last 3 frames in $\mathbf{Y}^{(0)}$ contain foreground movement (man), vectorize each video frame and arrange them as columns of data matrix $\mathbf{X} \in \mathbb{R}^{54338 \times 474}$. We set $n = 20$ and $\tau = 0.9$ and apply L1-IPCA to compute the $K = 5$ L1-PCs of the video sequence $\hat{\mathbf{Q}} \in \mathbb{R}^{54338 \times 5}$. The background of the i-th frame $\mathbf{x}_i = [\mathbf{X}]_{:,i}$ is obtained by projecting it onto the computed K L1-PCs as $\mathbf{x}_i^{(\text{back})} = \hat{\mathbf{Q}}\hat{\mathbf{Q}}^T\mathbf{x}_i$. Then, the foreground frame is obtained through background subtraction; that is $\mathbf{x}_i^{(\text{fore})} = \mathbf{x}_i - \mathbf{x}_i^{(\text{back})}$. In Figure4.12a, we present the 135-th frame of the processed video sequence. In addition, in Figure 4.12 we present the background extracted by (b) ISVD [62], (c) GRASTA [82], (d) PCP [19], (e) Online-RPCA via stochiastic optimization (OR-PCA) [74], (f) RPCA [29], (g) ReProCS [72], and (h) L1-IPCA (proposed). Foreground extracted by (i) ISVD [62], (j) GRASTA [82], (k) PCP [19], (l) OR-PCA [74], (m) RPCA [29], (n) ReProCS [72], and (o) L1-IPCA (proposed). We observe that background computed by ISVD [62] exhibits a

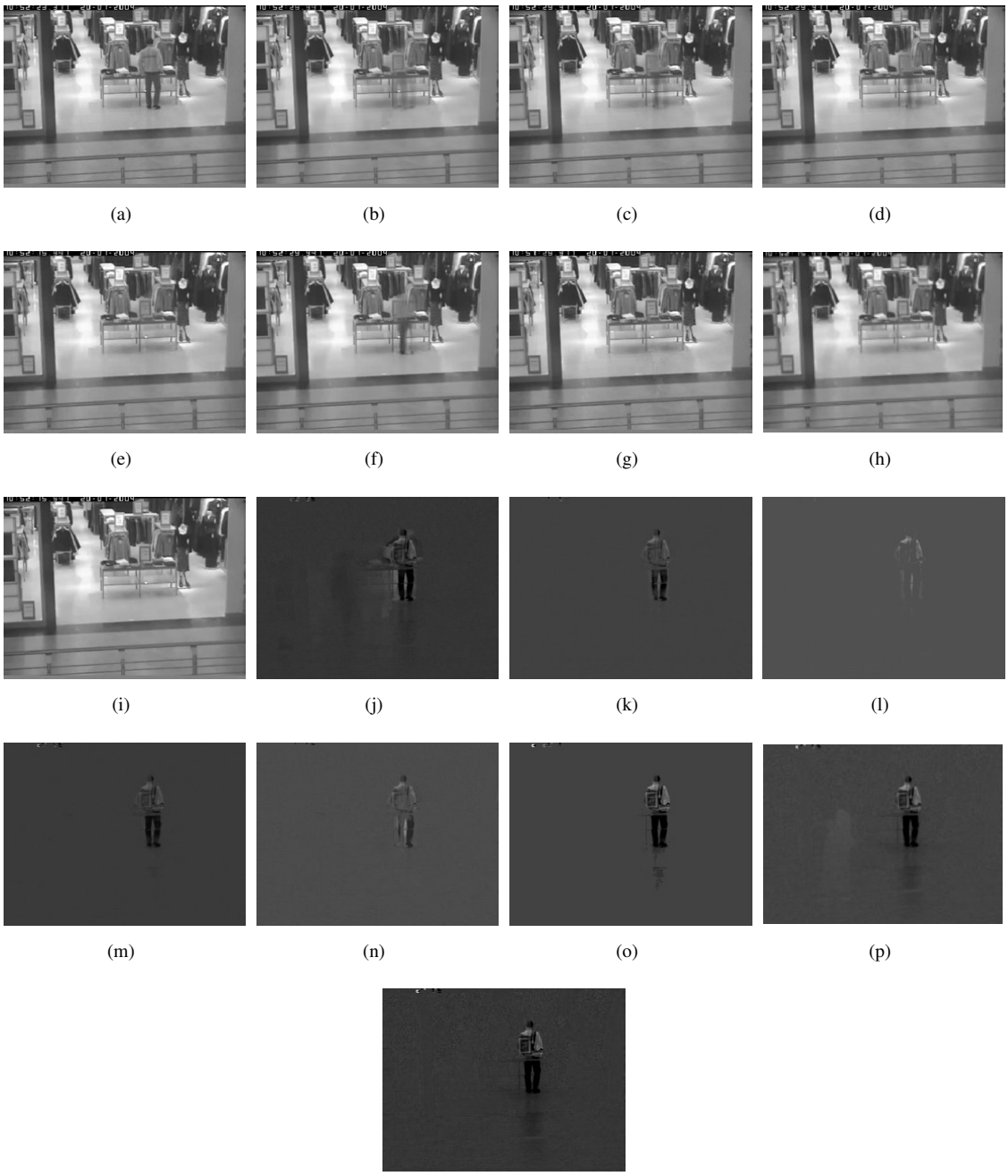


Figure 4.12: Video processing experiment – video 1. (a) Original frame. Background extracted by (b) ISVD [62], (c) GRASTA [82], (d) PCP [19], (e) OR-PCA [74], (f) RPCA [29], (g) ReProCS [72], (h) method of [47], and (i) L1-IPCA (proposed). Foreground extracted by (j) ISVD [62], (k) GRASTA [82], (l) PCP [19], (m) OR-PCA [74], (n) RPCA [29], (o) ReProCS [72], (p) method of [47], and (q) L1-IPCA (proposed).

non-negligible “ghostly” appearance of the walking man, whose blurred/inaccurate figure also appears in the foreground. GRASTA [82], PCP [19], and RPCA [29] exhibit similar performance with the smudged appearance of the man in the computed background. OR-PCA [74] performs clearly better than the previous methods. The method of [47] extracts a clean background but "ghostly" appearances of the ladies in the last frame are seen on the extracted foreground. The proposed L1-IPCA algorithm, together with ReProCS [72], demonstrate similarly high performance, obtaining the clean background and a foreground with a well defined outline of the man, together with his shadow.

Next, we obtain a surveillance video of the entrance lobby at INRIA labs in France from the same CAVIAR database [91]. We keep only the first $N = 295$ frames, vectorize them, and arrange them as columns of data matrix $\mathbf{X} \in \mathbb{R}^{54338\times 295}$. We set $n = 14$ and $\tau = 0.9$ and repeat the experiment to obtain the background and foreground of the 80-th frame of the video using $K = 3$ PCs. In Figure 4.13, we plot the performance of ISVD [62], GRASTA [82], PCP [19], OR-PCA [74], RPCA [29] and ReProCS [72] (background and extracted foreground). We observe that ISVD [62], GRASTA [82], PCP [19], and RPCA [29] demonstrate again similar performance as before –i.e., ghostly appearance of the man in the extracted background and his blurred figure in the foreground. The method of [47] extracts a clean background but "ghostly" appearances of the foreground movement in the last frame is seen on the extracted foreground. On the other hand, OR-PCA [74], ReProCS [72], and L1-IPCA obtain similarly clean background and well-defined foreground.

Finally, we operate on the “Curtain Video” [85]. We collect $N = 103$ frames of size 45 by 46 pixels, vectorize them, and arrange them as columns of data matrix $\mathbf{X} \in \mathbb{R}^{2520\times 103}$.

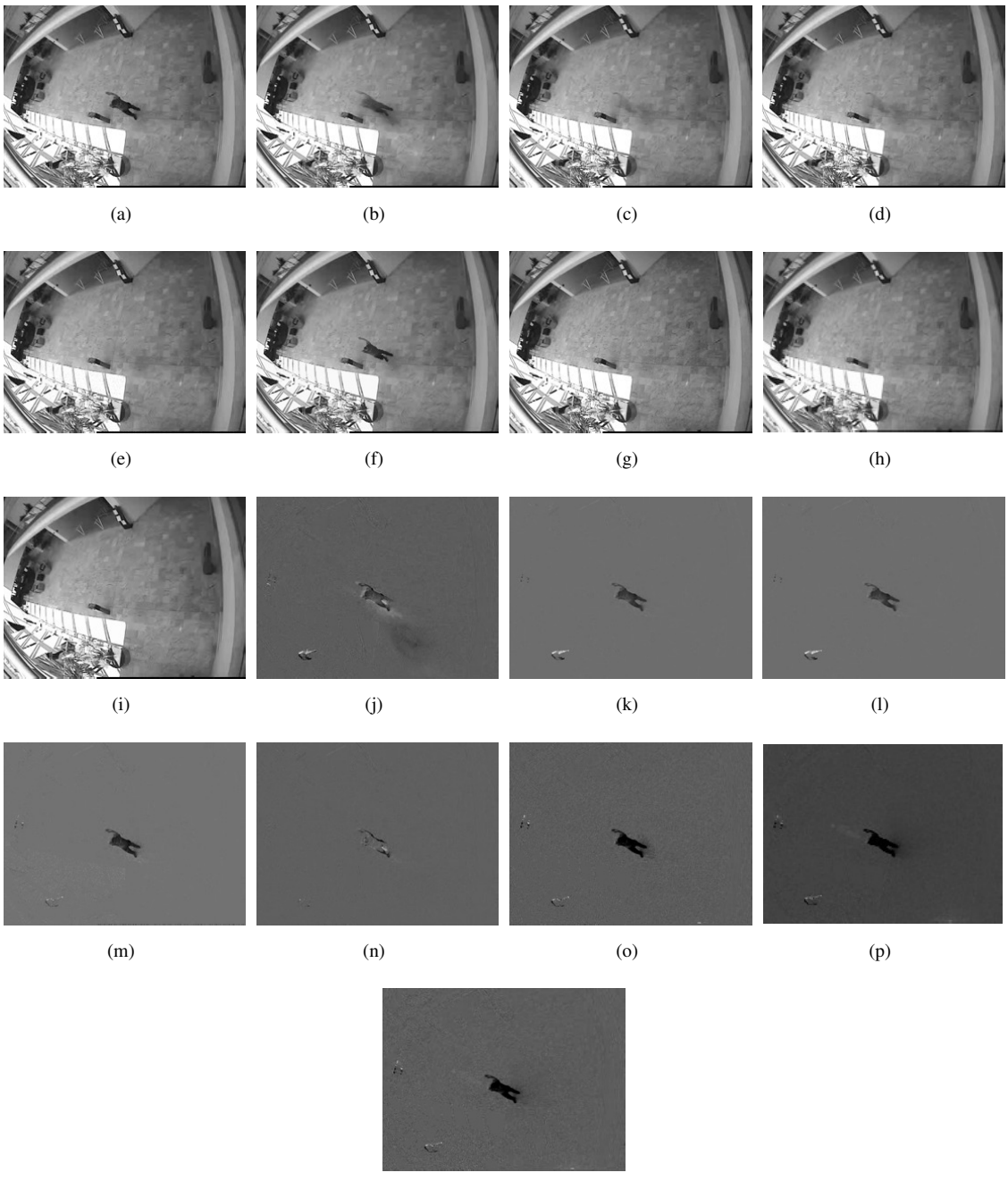


Figure 4.13: Video processing experiment – video 2. (a) Original frame. Background extracted by (b) ISVD [62], (c) GRASTA [82], (d) PCP [19], (e) OR-PCA [74], (f) RPCA [29], (g) ReProCS [72], (h) method of [47], and (i) L1-IPCA (proposed). Foreground extracted by (j) ISVD [62], (k) GRASTA [82], (l) PCP [19], (m) OR-PCA [74], (n) RPCA [29], (o) ReProCS [72], (p) method of [47], and (q) L1-IPCA (proposed).

We set $n = 5$ and $\tau = 0.999$ and repeat the above experiment, for $K = 2$. In Figure 4.14, we present the foreground and background of the 50-th frame as obtained by ISVD [62], GRASTA [82], PCP [19], OR-PCA [74], RPCA [29], and ReProCS [72]. We notice that this experiment poses some particular challenges: the man's shirt matches the background curtain color, the man (foreground) is stationary in many frames, and the curtain in the background of this video moves continuously leading to slow background changes. In Figure 4.14, we observe that the background frames obtained by ISVD [62], GRASTA [82], PCP [19], OR-PCA [74], and RPCA [29] retain the majority of the foreground (man); at the same time, the corresponding foreground frames do not capture the man clearly. the method of [47] has slight reminiscence of the man in the foreground and traces of the moving (background) curtain in its extracted foreground. On the other hand, ReProCS [72] obtains a cleaner background with slight presence of the man, while the proposed L1-IPCA algorithm obtains an entirely clean background. The foreground extracted by L1-IPCA contains some traces of the moving (background) curtain, which are not present in the foreground of ReProCS [72].

4.4 Direction-of-Arrival Estimation and Tracking

Going forward, an experiment on direction-of-arrival (DoA) estimation and tracking is performed. We consider uniform linear antenna array (ULA) of $D = 4$ antenna elements that capture $N = 70$ snapshots of an incoming signal of interest that arrives from angle $\phi = -40^\circ$ with respect to the broadside. The i-th down-converted and pulse-matched

snapshot takes the form

$$\mathbf{x}_i = b_i \mathbf{s}(\phi) + \mathbf{n}_i, \ i = 1, 2, \ldots, 70 \tag{4.2}$$

where $\mathbf{s}(\phi) = [1, e^{-j\pi sin(\phi)}, \ldots, e^{-j\pi sin(\phi)(D-1)}]^T$ is the array-response vector, b_i is i-th symbol (accounting for transmission energy and channel attenuation) with $b_i \in \{\pm\sqrt{\alpha}\}$, $\alpha = 10$, and n_i is complex AWGN from $\mathcal{CN}(0_4, I_4)$. The 70 snapshots are stacked as the columns of data matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N] \in \mathbb{C}^{4\times 70}$. We assume that snapshots 5 and 55 are unexpectedly corrupted by a jamming signal from DoA $\phi_o = 10°$, carrying a symbol from $\{\pm\sqrt{\beta}\}$, $\beta = 60$.

To estimate ϕ, the receiver operates as follows. First, $\mathbf{X}$ is realified as $\tilde{\mathbf{X}} = [\Re\{\mathbf{X}\}, \Im\{\mathbf{X}\}]^T \in \mathbb{R}^{8\times}$, where $\Re\{\cdot\}$ and $\Im\{\cdot\}$ return the real and imaginary parts of their arguments respectively. Next, we estimate the $K = 1$ PC of $\mathbf{X}$ by L1-IPCA with parameters $n = 20$, and $\tau = 0.9$. For approximate L1-PC $\hat{\mathbf{q}}_i$, we compute the L1-PCA-based MUSIC-type [52] spectrum

$$P_i(\theta) = \frac{1}{\| (\mathbf{I}_8 - \hat{\mathbf{q}}_i \hat{\mathbf{q}}_i^T)\tilde{\mathbf{s}}(\theta)\|_2}, \tag{4.3}$$

for θ in $\Theta = \{\frac{-\pi}{2}, \frac{-\pi}{2} + \Delta, \ldots, \frac{\pi}{2} - \Delta\}$, for arbitrarily small step $\Delta > 0$, and $\tilde{\mathbf{s}}(\theta) = [\Re\{\mathbf{s}(\theta)\}^T, \Im\{\mathbf{s}(\theta)\}^T]^T$. Similar to [52], the i-th estimate of ϕ is given by

$$\hat{\phi}_i = \underset{\theta\in\Theta}{\text{argmax}} \ P_i(\theta). \tag{4.4}$$

We carry out DoA estimation using the PC obtained by batch SVD, ISVD, GRASTA, and OR-PCA. In Figure 4.15, we plot an instance of $P_{70}(\theta)$ for all 5 methods. We observe

that the L2-based methods SVD and ISVD are misled by the jamming signal and point towards $\phi_o = 10^\circ$. On the other hand, the robust GRASTA, OR-PCA, and L1-APCA (most emphatically) point towards the correct DoA $\phi = -40^\circ$.

Next, by keeping all the parameters the same, we set $\alpha = 1$ and $\beta = 33$ (leading to SNR (source) = 0 dB and SNR (jammer) = 15dB) and re-run the DoA estimation experiment to plot in Figure 4.16 an instance of $P_{70}(\theta)$ for all 5 methods. Again, we observe that SVD and ISVD are misled by the jammer. (they have two peaks, one at the DoA of source and the other at DoA of jammer, however the peak at jammer is higher and hence considered more important). GRASTA, OR-PCA and L1-IPCA point correctly at the source DoA.

In the sequel, we increase $N = 200$ and steer our focus towards DoA tracking. We consider that in the first 90 snapshots the signal of interest arrives from DoA $\phi_1 = -40^\circ$. In the latter 110 snapshots, the signal of interest arrives from DoA $\phi_2 = -35^\circ$ (signal subspace change). We consider a jammer at $\phi_o = -60^\circ$ corrupting snapshots 5, 55, and 135. We run L1-APCA with parameters $n = 20$, $\tau = 0.9$, $\rho = 0.8$, and $q = 0.9n$ to track the DoA of the signal of interest. In Figure 4.17, we plot the root-mean-squared-error (RMSE) (average over 2000 independent realizations) versus adaptation index i, calculated as

$$\text{RMSE}_i = \sqrt{\frac{1}{2000} \sum_{m=1}^{2000} |\hat{\phi}_i^{(m)} - \phi|^2}, \tag{4.5}$$

where $\hat{\phi}_i^{(m)}$ is the DoA estimation at the i-th adaptation of the m-th realization. In (4.5), $\phi = \phi_1$ for $i \leq 90$ and $\phi = \phi_2$ for $i > 90$. We observe that SVD and ISVD are mislead by the jammers and attain high RMSE for every i. GRASTA, OR-PCA, and the proposed L1-APCA exhibit both robustness against jamming and the ability to adapt quickly to changes

of the signal DoA. L1-APCA attains consistently superior RMSE performance.

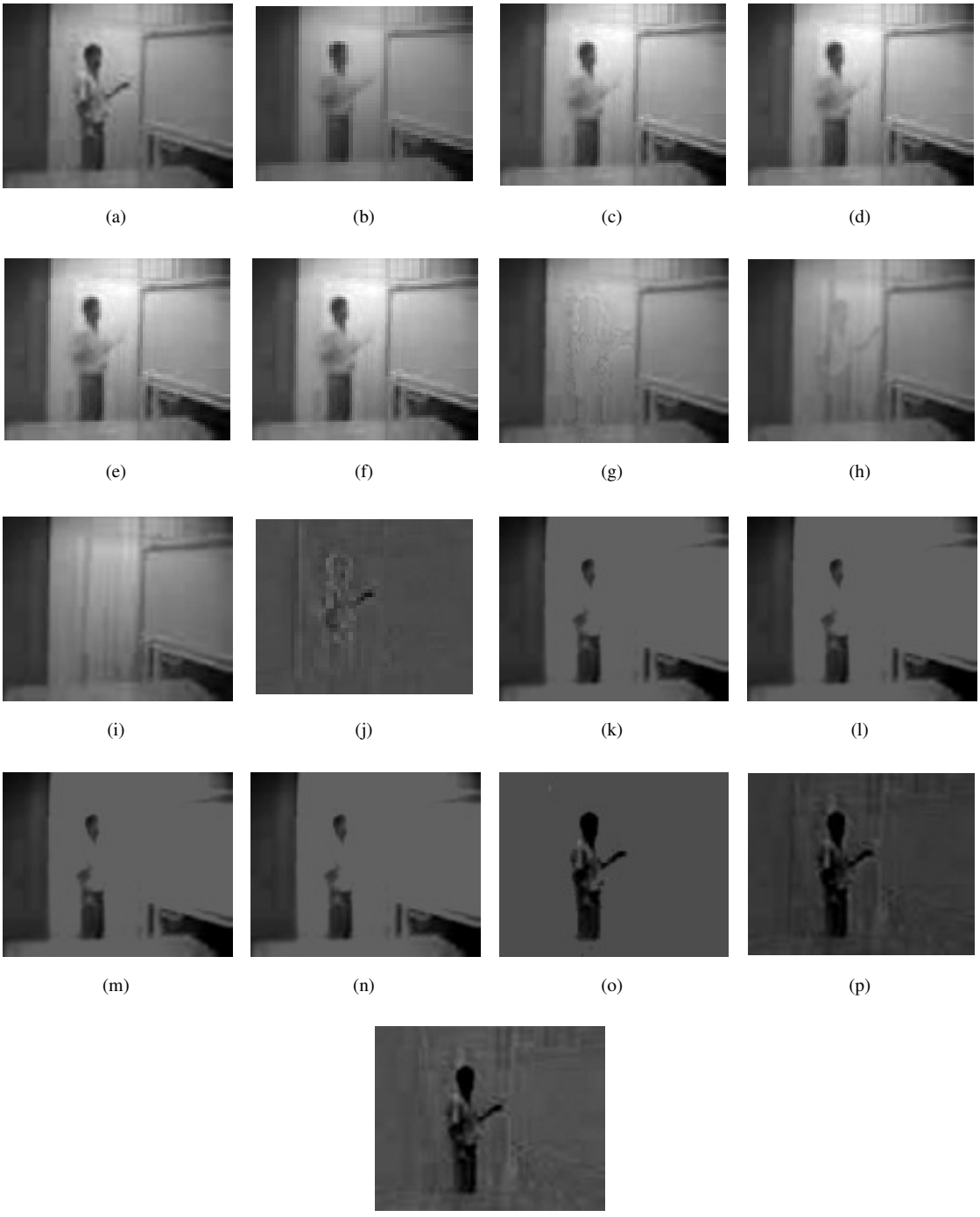


Figure 4.14: Video processing experiment – video 3. (a) Original frame. Background extracted by (b) ISVD [62], (c) GRASTA [82], (d) PCP [19], (e) OR-PCA [74], (f) RPCA [29], (g) ReProCS [72], (h) method of [47], and (i) L1-IPCA (proposed). Foreground extracted by (j) ISVD [62], (k) GRASTA [82], (l) PCP [19], (m) OR-PCA [74], (n) RPCA [29], (o) ReProCS [72], (p) method of [47], and (q) L1-IPCA (proposed).

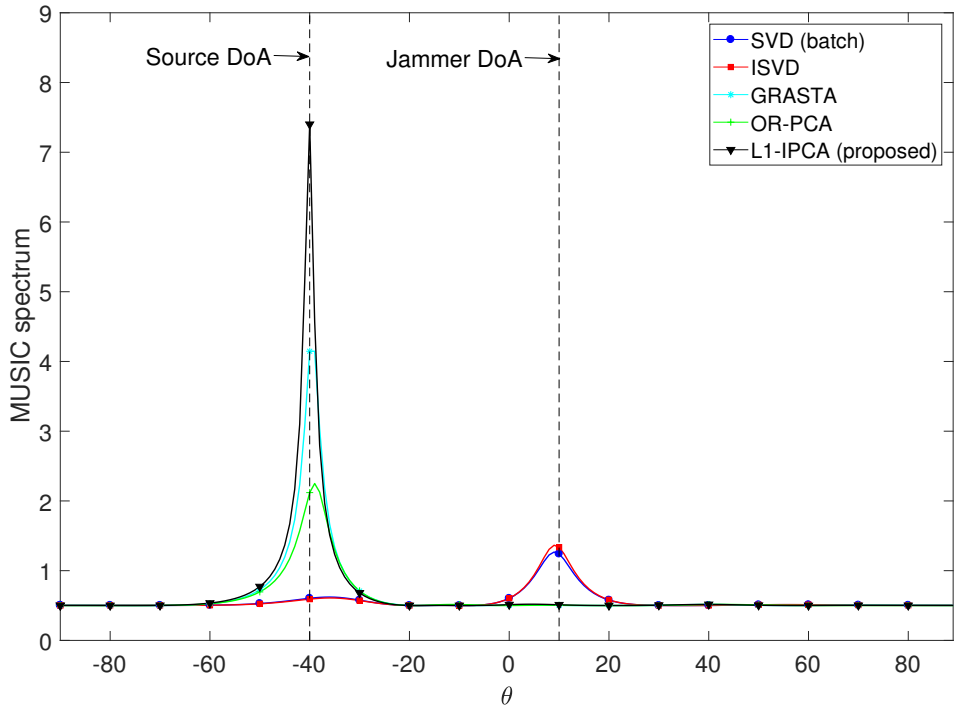


Figure 4.15: DoA estimation experiment. DoA estimation spectrum $P_{70}(\theta)$. $N = 70$, $D = 4$, $K = 1$. $\phi = -40°$, $\phi_o = 10°$, $\alpha = 10$, $\beta = 60$. $n = 20$, $\tau = 0.9$. Jamming at $\mathbf{x}_5$ and $\mathbf{x}_{55}$.

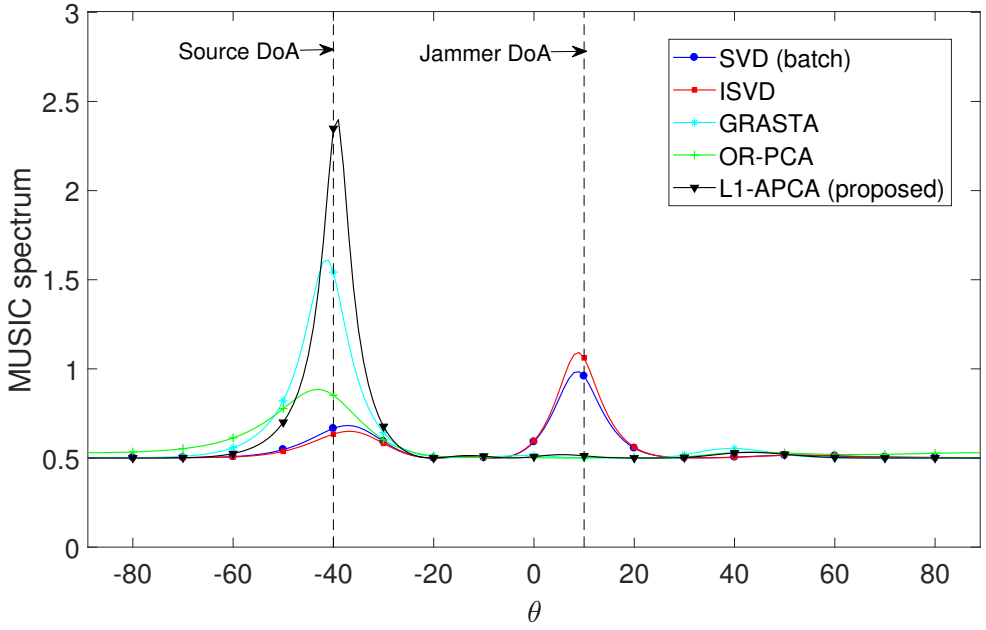


Figure 4.16: DoA estimation experiment. DoA estimation spectrum $P_{70}(\theta)$. $N = 70$, $D = 4$, $K = 1$. $\phi = -40°$, $\phi_o = 10°$, $\alpha = 1$, $\beta = 33$. $n = 20$, $\tau = 0.9$. Jamming at $\mathbf{x}_5$ and $\mathbf{x}_{55}$.

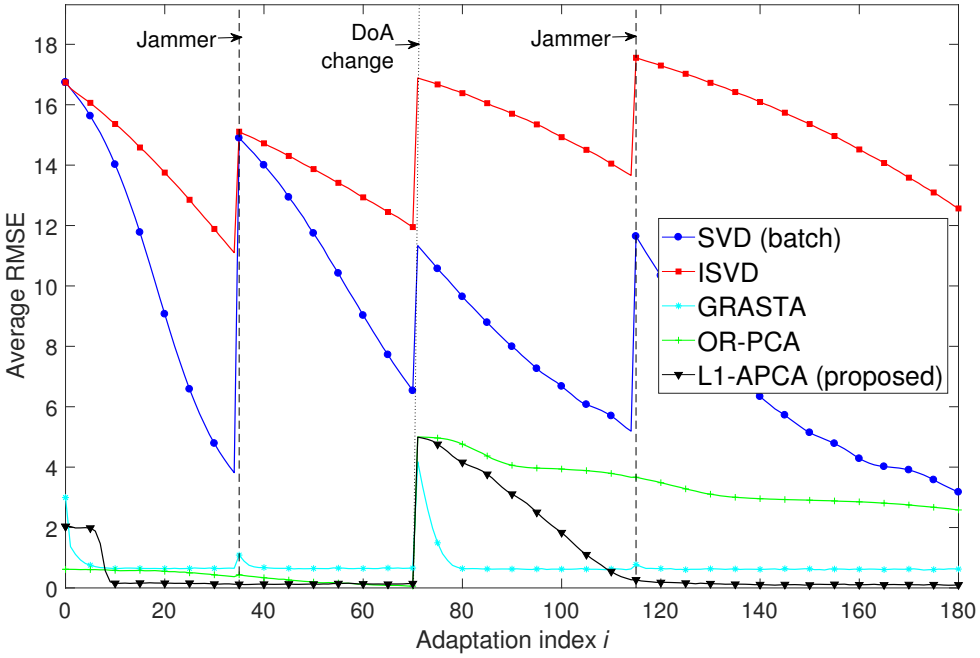


Figure 4.17: DoA tracking experiment. RMSE performance versus adaptation index i. $N = 200$, $D = 4$, $K = 1$. $\phi_1 = -40°$, $\phi_2 = -35°$, $\phi_o = -60°$, $\alpha = 10$, $\beta = 60$. $n = 20$, $\tau = 0.9$, $\rho = 0.8$, $q = 0.9n$. Jamming at $\mathbf{x}_5$, $\mathbf{x}_{55}$, and $\mathbf{x}_{135}$.

Chapter 5

Quality of Initialization and Parameter Tuning

The quality of initialization memory batch $\mathbf{Y}_0$ is important as the reliability of incoming data points is evaluated based on the PCs obtained from $\mathbf{Y}_0$. If $\mathbf{Y}_0$ is sufficiently outlier-corrupted, the the PCs obtained $\hat{\mathbf{Q}}_0$ might be incorrect and thus new nominal data points will be discarded due to their low reliability with respect to such a $\hat{\mathbf{Q}}_0$. In L1-APCA, because of the threshold decrease ratio ρ and q most recent measurements preserved in memory batch, the algorithm recovers from an incorrect $\hat{\mathbf{Q}}_0$ as it processes new nominal data points. L1-IPCA however, does now have such a mechanism and thus a sufficiently clean memory batch initialization is required.

In the sequel we say a few things about tuning important parameters of our algorithms. In L1-IPCA: the memory batch size n could be chosen to be sufficiently large such that the ratio of number of nominal points to number of outlier-corrupted points is high (ideally, close to 1). Threshold τ could be chosen close to 1. E.g., in video/image processing experiments we set τ to a value close to 1 and obtain good performance. However, if the initial memory batch is not sufficiently clean and/or the noise in the data measurements is high, then a high τ would lead to incorrect solutions. If the confidence of the initial solution $\hat{\mathbf{Q}}_0$ is less or the noise is high, then a low τ would be a correct initialization.

In L1-APCA: n is chosen as in L1-IPCA. τ_{max} could be set close to 1. If $\mathbf{Y}_0$ contains many corrupted measurements, the threshold drops due to the use of threshold degradation. The threshold decrease ration ρ could be set to a lower value if the expected change in subspace is drastic and/or if faster adaptation is required during subspace change. However, if ρ is set too small, then when an outlier occurs, the threshold value used for reliability check will be lowered enough for any successive outliers that occur in sequence to enter the memory batch and thus lead to incorrect solutions. E.g., in our synthetic data experiments, we set $\rho = 0.35$ (low value) as the angle of subspace change is 32.43° and because successive outliers do not occur. Number of recent points protected, q could generally be set to about 25% - 75% of n for good performance. Greater q means more number of recent points preserved in memory batch, leading to faster adaptation during subspace change. However, if a burst of outliers (multiple outliers in sequence) occur, then a large q will lead to incorrect solutions and in such environments, we could use a mid-range/low q.

Chapter 6

Conclusions

The use of traditional batch L1-PCA algorithms may be prohibitive in big-data (large N) or heavy-data (large D) applications due to their high computational costs, although they are robust against outliers. Moreover, in streaming/tracking applications, computing L1-PCs from scratch at the arrival of every new data-point is unfeasible, again due to increasing cost. The algorithms presented in this thesis aim at performing L1-PCA in such scenarios and thus may connect the dots between fast, low complexity and outlier-resistant L1-PCA in "*big-data*" or "*big streaming-data*" applications.

An algorithmic framework for incremental and adaptive L1-PCA is proposed. The first algorithm (L1-IPCA) updates L1-PCA incrementally with low computational cost (linear in the number of data points), maintaining sturdy resistance against outliers. Its efficacy is verified by experimental studies on subspace estimation on synthetic data, image conditioning, video processing, and DoA estimation. The second algorithm (L1-APCA), deriving by L1-IPCA after two modifications, is capable of adapting the L1-PCA solution to changes in the nominal signal subspace, while remaining robust against outliers. Its potency is verified by experimental studies on dynamic subspace tracking on synthetic data, image conditioning, video processing, and DoA tracking. Additionally, our experimental studies verify

the computational efficiency, outlier resistance, and updation/adaptation capabilities of the proposed algorithms compared to state-of-the-art alternatives.

Chapter 7

Future Work

The proposed algorithms operate on a constant-size memory batch of n columns. This leads to lower computational cost as only n points are processed at any i-th iteration, no matter how big i might be. Going further, PC adaptation/update could be performed using only the incoming entry $\mathbf{x}^{(in)}$ and the previous solution $\hat{\mathbf{Q}}_{i-1}$.

The second algorithm, i.e., L1-APCA can benefit from the following feature – Use of secondary memory: As explained earlier, the proposed algorithms evaluate the reliability of each incoming point by measuring its angular proximity to the currently computed L1-PCs. If this proximity is below a threshold τ, then the incoming point is disregarded as a possible outlier. In the case of dynamically changing signal subspace, a data point from the new/changed signal subspace may seem as an outlier when evaluated by the L1-PCs computed on data from the old signal subspace. The proposed algorithms would likely disregard such a point as outlier (although threshold degradation/adjustment is used in L1-APCA would eventually allow data points from the new signal-subspace into the memory batch for PC-adaptation, the first few points from a new signal subspace are discarded most of the time), missing the opportunity of using it to track the subspace change. A new version L1-APCA algorithm may keep a secondary memory of limited size $m > 0$ where

it stores incoming points that fail the L1-reliability check and use it again for processing when the threshold value is suitable degraded. The use of secondary memory in L1-APCA would certainly speed up the subspace adaptation whenever the signal-subspace changes.

In both L1-IPCA and L1-APCA, a forgetting factor could be used to forget older measurements in the memory batch. Such a forgetting factor could save our algorithms from incorrect solutions due to a bad initial memory batch (that may contain many outliers). A weighting factor could be used to weight the measurements in memory batch so that the contribution of each measurement for PC-update/adaptation depends on its weight. Older measurements could be weighted less (i.e., forgotten by some forgetting factor) compared to newer ones.

Bibliography

[1] I. T. Jolliffe, *Principal Component Analysis.* New York, NY. Springer, 1986.

[2] M. E. Wall, A. Rechtsteiner, and L. M. Rocha, "Singular value decomposition and principal component analysis," in *Pract. App. Micr. Data Anal.* Springer, 2003, pp. 91–109.

[3] K. Pearson, "On lines and planes of closest fit to systems of points in space," *Philosoph. Mag.*, vol. 2, pp. 559–572, 1901.

[4] G. Shaw and D. Manolakis, "Signal processing for hyperspectral image exploitation," *IEEE Signal Process. Mag.*, vol. 19, pp. 12–16, Feb. 2002.

[5] Q. Du and J. E. Fowler, "Hyperspectral image compression using jpeg2000 and principal component analysis," *IEEE Geosci. Remote Sens. Lett.*, vol. 4, pp. 201–205, May 2007.

[6] O. Edfors, M. Sandell, J.-J. Van de Beek, S. K. Wilson, and P. O. Borjesson, "OFDM channel estimation by singular value decomposition," *IEEE Trans. Commun.*, vol. 46, pp. 931–939, 1998.

[7] H. Q. Ngo and E. G. Larsson, "EVD-based channel estimation in multicell multiuser mimo systems with very large antenna arrays," in *IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP),* Kyoto, Japan, Mar. 2012, pp. 3249–3252.

[8] M. B. Christopher, *Pattern Recognition and Machine Learning*. New York, NY: Springer-Verlag, 2016.

[9] C. Ding and X. He, "K-means clustering via principal component analysis," in *Proc. ACM Int. Conf. Mach. Learn.,* Alberta, Canada, Jul. 2004, p. 29.

[10] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification, 2nd ed.* New York, NY: John Wiley & Sons, 2012.

[11] J. Yang, D. Zhang, A. F. Frangi, and J. Y. Yang, "Two-dimensional PCA: a new approach to appearance-based face representation and recognition," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 26, pp. 131–137, Jan. 2004.

[12] S. Sanei and J. A. Chambers, *EEG Signal Processing*. John Wiley & Sons, 2013.

[13] F. Castells, P. Laguna, L. Sörnmo, A. Bollmann, and J. M. Roig, "Principal component analysis in ECG signal processing," *EURASIP J. App. Signal Process.*, vol. 2007, pp. 98–98, Dec. 2007.

[14] K. Y. Yeung and W. L. Ruzzo, "Principal component analysis for clustering gene expression data," *Bioinformatics*, vol. 17, pp. 763–774, Sep. 2001.

[15] V. Gupta, R. Singh, G. Singh, R. Singh, and H. Singh, "An introduction to principal component analysis and its importance in biomedical signal processing," in *Proc. Int. Conf. Life Sci. Tech.*, vol. 3, 2011, pp. 29–33.

[16] H. Hotelling, "Analysis of a complex of statistical variables into principal components," *J. Ed. Psych.*, vol. 24, pp. 417–441, 1933.

[17] C. Eckart and G. Young, "The approximation of one matrix by another of lower rank," *Psychometrika*, vol. 1, pp. 211–218, 1936.

[18] V. Barnett and T. Lewis, *Outliers in statistical data*. New York, NY: Wiley, 1994.

[19] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, pp. 1–39, May 2011.

[20] G. Mateos and G. B. Giannakis, "Robust PCA as bilinear decomposition with outlier-sparsity regularization," *IEEE Trans. Signal Process.*, vol. 60, pp. 5176–5190, Oct. 2012.

[21] M. Chen, A. Ganesh, Z. Lin, Y. Ma, J. Wright, and L. Wu, "Fast convex optimization algorithms for exact recovery of a corrupted low-rank matrix," *J. Coord. Sci. Lab. Rep. no. UILU-ENG-09-2214*, 2009.

[22] V. Chandrasekaran, S. Sanghavi, P. A. Parrilo, and A. S. Willsky, "Sparse and low-rank matrix decompositions," *Proc. IFAC Symp. Syst. Ident.*, vol. 42, pp. 1493–1498, 2009.

[23] H. Xu, C. Caramanis, and S. Sanghavi, "Robust PCA via outlier pursuit," in *Proc. Adv. in Neur. Info. Process. Syst. (NIPS),* Vancouver, Canada, Dec. 2010, pp. 2496–2504.

[24] J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma, "Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization," in *Proc. Advan. Neural Inf. Process. Syst. (NIPS),* Vancouver, Canada, 2009, pp. 2080–2088.

[25] V. Chandrasekaran, S. Sanghavi, P. A. Parrilo, and A. S. Willsky, "Rank-sparsity incoherence for matrix decomposition," *J. SIAM Optim.*, vol. 21, pp. 572–596, Apr. 2011.

[26] B. Recht, M. Fazel, and P. A. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," *J. SIAM Rev.*

[27] C. Ding, D. Zhou, X. He, and H. Zha, "R1-PCA: rotational invariant L1-norm principal component analysis for robust subspace factorization," in *Proc. 23rd Int. Conf. Mach. Lear.*, Jun., pp. 281–288.

[28] F. De La Torre and M. J. Black, "A framework for robust subspace learning," *Int. J. Comput. Vis.*, vol. 54, pp. 117–142, Aug. 2003.

[29] F. De la Torre and M. J. Black, "Robust principal component analysis for computer vision," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV),* Venice, Italy, Jul. 2001, pp. 362–369.

[30] X. Yi, D. Park, Y. Chen, and C. Caramanis, "Fast algorithms for robust PCA via gradient descent," in *Proc. Adv. Neu. Info. Process. Syst. (NIPS),* Barcelona, Spain, Dec. 2016, pp. 4152–4160.

[31] Q. Ke and T. Kanade, "Robust L1 factorization in the presence of outliers and missing data by alternative convex programming," in *Proc. IEEE Conf. Comput. Vision Patt. Recogn. (CVPR),* San Diego, CA, Jun. 2005, pp. 739–746.

[32] J. P. Brooks, J. H. Dulá, and E. L. Boone, "A pure L1-norm principal component analysis," *Elsevier Comput. Stat. & Data Anal.*, vol. 61, pp. 83–98, May 2013.

[33] N. Tsagkarakis, P. P. Markopoulos, and D. A. Pados, "On the L1-norm approximation of a matrix by another of lower rank," in *Proc. IEEE Conf. Mach. Learn. and App.,* Anaheim, CA, Dec. 2016, pp. 768–773.

[34] P. P. Markopoulos, G. N. Karystinos, and D. A. Pados, "Optimal algorithms for L1-subspace signal processing," *IEEE Trans. Signal Process.*, vol. 62, pp. 5046–5058, Oct. 2014.

[35] P. P. Markopoulos, S. Kundu, S. Chamadia, and D. Pados, "Efficient L1-norm principal-component analysis via bit flipping," *IEEE Trans. Signal Process.*, vol. 65, pp. 4252–4264, Aug. 2017.

[36] N. Kwak, "Principal component analysis based on L1-norm maximization," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 30, pp. 1672–1680, Sep. 2008.

[37] F. Nie, H. Huang, C. Ding, D. Luo, and H. Wang, "Robust principal component analysis with non-greedy L1-norm maximization," in *Proc. Int. Joint Conf. Art. Intell. (IJCAI),* Barcelona, Spain, Jul. 2011, pp. 1433–1438.

[38] M. McCoy and J. A. Tropp, "Two proposals for robust PCA using semidefinite programming," *Electron. J. Statist.*, vol. 5, pp. 1123–1160, Jun. 2011.

[39] P. P. Markopoulos, D. A. Pados, G. N. Karystinos, and M. Langberg, "L1-norm principal-component analysis in L2-norm-reduced-rank data subspaces," in *Proc. SPIE,* Anaheim, CA, vol. 10211, May. 2017, pp. 04:1–04:10.

[40] P. P. Markopoulos. L1-PCA code repository. [Online]. Available: https://people.rit.edu/pxmeee/soft.

[41] N. Tsagkarakis, P. P. Markopoulos, G. Sklivanitis, and D. A. Pados, "L1-norm principal-component analysis of complex data," *IEEE Trans. Signal Process.*, Apr. 2018.

[42] S. Chamadia and D. A. Pados, "Optimal sparse L1-norm principal-component analysis," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP),* New Orleans, LA, Mar. 2017, pp. 2686–2690.

[43] P. P. Markopoulos, S. Kundu, and D. A. Pados, "L1-fusion: Robust linear-time image recovery from few severely corrupted copies," in *Proc. IEEE Int. Conf. Image Process. (ICIP),* Quebec, Canada, Sep. 2015, pp. 1225–1229.

[44] M. Johnson and A. Savakis, "Fast L1-eigenfaces for robust face recogntion," in *Proc. IEEE West. New York Image Signal Process. Workshop (WNYISPW),* Rochester, NY, Nov. 2014, pp. 1–5.

[45] F. Maritato, Y. Liu, S. Colonnese, and D. A. Pados, "Cloud-assisted individual L1-PCA face recognition using wavelet-domain compressed images," in *Proc. IEEE Euro. Workshop Vis. Info. Process. (EUVIP),* Marseille, France, Oct. 2016, pp. 1–6.

[46] ——, "Face recognition with L1-norm subspaces," in *Proc. SPIE,* Baltimore, MD, vol. 9857, May 2016, pp. 0L:1–0L:8.

[47] M. Pierantozzi, Y. Liu, D. A. Pados, and S. Colonnese, "Video background tracking and foreground extraction via L1-subspace updates," in *Proc. SPIE,* Baltimore, MD, vol. 9857, May 2016, pp. 08:1–08:16.

[48] Y. Liu and D. A. Pados, "Compressed-sensed-domain L1-PCA video surveillance," *IEEE Trans. Multimed.*, vol. 18, pp. 351–363, Mar. 2016.

[49] D. G. Chachlakis, P. P. Markopoulos, R. J. Muchhala, and A. Savakis, "Visual tracking with L1-Grassmann manifold modeling," in *Proc. SPIE,* Anaheim, CA, vol. 10211, Apr. 2017, pp. 02:1–02:10.

[50] P. P. Markopoulos, "Reduced-rank filtering on L1-norm subspaces," in *Proc. IEEE Workshop Sens. Array Multichannel Signal Process. (SAM),* Rio de Janeiro, Brazil, Jul. 2016, pp. 1–5.

[51] N. Tsagkarakis, P. P. Markopoulos, and D. A. Pados, "Direction finding by complex L1-principal-component analysis," in *Proc. IEEE Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC),* Stockholm, Sweden, Jun. 2015, pp. 475–479.

[52] P. P. Markopoulos, N. Tsagkarakis, D. A. Pados, and G. N. Karystinos, "Direction-of-arrival estimation by L1-norm principal components," in *Proc. IEEE Int. Symp. Phased Array Syst. Tech. (PAST 2016),* Waltham, MA, Oct. 2016, pp. 1–6.

[53] P. P. Markopoulos and F. Ahmad, "Indoor human motion classification by L1-norm subspaces of micro-doppler signatures," in *Proc. IEEE Radar Conf.,* Seattle, WA, May 2017, pp. 1807–1810.

[54] Y. Liu, D. A. Pados, S. N. Batalama, and M. J. Medley, "Iterative re-weighted L1-norm principal-component analysis," in *Proc. IEEE Asilomar Conf. Signals, Syst., Comput.,* Pacific Grove, CA, 2017, pp. 425–429.

[55] P. P. Markopoulos, D. G. Chachlakis, and E. E. Papalexakis, "The exact solution to rank-1 L1-norm TUCKER2 decomposition," *J. IEEE Signal Process. Lett.*, vol. 25, pp. 511–515, Jan. 2018.

[56] D. G. Chachlakis and P. P. Markopoulos, "Robust decomposition of 3-way tensors based on L1-norm," in *Proc. SPIE Def. Comm. Sensing (SPIE DCS),* Orlando, FL, vol. 10658, May 2018, pp. 07:1 – 07:15.

[57] N. Vaswani, T. Bouwmans, S. Javed, and P. Narayanamurthy, "Robust PCA and robust subspace tracking," *arXiv preprint arXiv:1711.09492*, 2017.

[58] F. Shang, J. Cheng, Y. Liu, Z.-Q. Luo, and Z. Lin, "Bilinear factor matrix norm minimization for robust PCA: Algorithms and applications," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 14, Sep. 2017.

[59] C. Wang, Y. Wang, Z. Lin, A. L. Yuille, and W. Gao, "Robust estimation of 3D human poses from a single image," in *Proc. IEEE Conf. Comput. Vis. Patt. Recogn. (CVPR),* Columbus, OH, Jun. 2014, pp. 2361–2368.

[60] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld, "Face recognition: A literature survey," *J. ACM Comput. Surveys (CSUR)*, vol. 35, pp. 399–458, Dec. 2003.

[61] P. Stoica and K. C. Sharman, "Maximum likelihood methods for direction-of-arrival estimation," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 38, pp. 1132–1143, Jul. 1990.

[62] J. R. Bunch and C. P. Nielsen, "Updating the singular value decomposition," *Numerische Mathematik*, vol. 31, pp. 111–129, 1978.

[63] L. Balzano, R. Nowak, and B. Recht, "Online identification and tracking of subspaces from highly incomplete information," in *Proc. IEEE Allerton Conf. Commun. Control Comput.,* Allerton, IL, Oct. 2010, pp. 704–711.

[64] S. Chandrasekaran, B. Manjunath, Y. F. Wang, J. Winkeler, and H. Zhang, "An Eigenspace update algorithm for image analysis," *Elsevier Graphic. Mod. Imag. Process.*, vol. 59, pp. 321–332, Sep. 1997.

[65] M. Brand, "Incremental singular value decomposition of uncertain data with missing values," in *Proc. Euro. Conf. Comput. Vis. Berlin, Heidelberg*. Springer, May 2002, pp. 707–720.

[66] ——, "Fast low-rank modifications of the thin singular value decomposition," *J. Lin. Alg. App.*, vol. 415, pp. 20–30, 2006.

[67] P. Comon and G. H. Golub, "Tracking a few extreme singular values and vectors in signal processing," *Proc. IEEE*, vol. 78, pp. 1327–1343, 1990.

[68] T. Bouwmans and E. H. Zahzah, "Robust PCA via principal component pursuit: A review for a comparative evaluation in video surveillance," *Elsevier Comput. Vis. Image Underst.*, vol. 122, pp. 22–34, May 2014.

[69] Y. Li, "On incremental and robust subspace learning," *Elsevier J. Patt. Recogn.*, vol. 37, pp. 1509–1518, Jul. 2004.

[70] N. Vaswani and P. Narayanamurthy, "Static and dynamic robust PCA via low-rank + sparse matrix decomposition: A review," *arXiv preprint arXiv:1803.00651*, Mar. 2018.

[71] J. Zhan, B. Lois, H. Guo, and N. Vaswani, "Online (and offline) robust PCA: Novel algorithms and performance guarantees," in *Proc. Art. Intell. Stat.,* Cadiz, Spain, May 2016, pp. 1488–1496.

[72] H. Guo, C. Qiu, and N. Vaswani, "An online algorithm for separating sparse and low-dimensional signal sequences from their sum," *IEEE Trans. Signal Process.*, pp. 4284–4297, Aug. 2014.

[73] J. Feng, H. Xu, S. Mannor, and S. Yan, "Online PCA for contaminated data," in *Proc. Adv. Neu. Info. Process. Syst. (NIPS),* Lake Tahoe, NV, 2013, pp. 764–772.

[74] J. Feng, H. Xu, and S. Yan, "Online robust PCA via stochastic optimization," in *Proc. Adv. Neu. Info. Process. Syst. (NIPS),* Lake Tahoe, NV, Dec. 2013, pp. 404–412.

[75] M. Mardani, G. Mateos, and G. B. Giannakis, "Dynamic anomalography: Tracking network anomalies via sparsity and low rank," *IEEE J. Select. Topics Signal Process.*, vol. 7, pp. 50–66, Feb. 2013.

[76] ——, "Recovery of low-rank plus compressed sparse matrices with application to unveiling traffic anomalies," *IEEE Trans. Info. Theory*, vol. 59, pp. 5186–5205, Aug. 2013.

[77] R. Otazo, E. Candès, and D. K. Sodickson, "Low-rank plus sparse matrix decomposition for accelerated dynamic MRI with separation of background and dynamic components," *J. Magn. Reson. Med.*, vol. 73, pp. 1125–1136, Mar. 2015.

[78] P. Narayanamurthy and N. Vaswani, "Provable dynamic robust PCA or robust subspace tracking," *arXiv preprint arXiv:1705.08948*, 2017.

[79] P. Netrapalli, U. Niranjan, S. Sanghavi, A. Anandkumar, and P. Jain, "Non-convex robust PCA," in *Proc. Adv. Neu. Info. Process. Syst. (NIPS),* Montreal, Canada, Dec. 2014, pp. 1107–1115.

[80] P. Rodriguez and B. Wohlberg, "Incremental principal component pursuit for video background modeling," *J. Math. Imag. Vis.*, vol. 55, pp. 1–18, May 2016.

[81] Y. Li, L. Xu, J. Morphett, and R. Jacobs, "An integrated algorithm of incremental and robust PCA," in *Proc. IEEE Int. Conf. Image Process. (ICIP),* Barcelona, Spain, Sep. 2003, pp. 245–248.

[82] J. He, L. Balzano, and A. Szlam, "Incremental gradient on the grassmannian for online foreground and background separation in subsampled video," in *Proc. IEEE Conf. Comput. Vis. Patt. Recog. (CVPR),*, Jun. 2012, pp. 1568–1575.

[83] J. He, L. Balzano, and J. Lui, "Online robust subspace tracking from partial information," *arXiv preprint arXiv:1109.3827*, 2011.

[84] A. Sobral, T. Bouwmans, and E. H. Zahzah, "LRS library: Low-rank and sparse tools for background modeling and subtraction in videos," in *Robust Low-Rank and Sparse Matrix Decomposition: Applications in Image and Video Processing*. CRC Press, 2015.

[85] H. Guo, C. Qiu, and N. Vaswani. Practical recursive projected compressive sensing. [Online]. Available: http://www.ece.iastate.edu/~hanguo/PracReProCS.

[86] J. He, L. Balzano, and J. Lui. GRASTA. [Online]. Available: https://sites.google.com/site/hejunzz/grasta.

[87] R. Basri and D. Jacobs, "Lambertian reflectance and linear subspaces," in *Proc. Eighth Int. Conf. Comput. Vis. (ICCV)*, vol. 2. IEEE, 2001, pp. 383–390.

[88] Psychological image collection at Stirling (PICS). [Online]. Available: http://pics.stir.ac.uk/.

[89] C. Rother, V. Kolmogorov, and A. Blake, "GrabCut: Interactive foreground extraction using iterated graph cuts," in *ACM Trans. Graphics*, vol. 23, Aug. 2004, pp. 309–314.

[90] B. Lei and L. Q. Xu, "Real-time outdoor video surveillance with robust foreground extraction and object tracking via multi-state transition management," *Elsevier J. Patt. Recogn. Lett.*, vol. 27, pp. 1816–1825, Nov. 2006.

[91] Context aware vision using image-based active recognition (CAVIAR). [Online]. Available: http://homepages.inf.ed.ac.uk/rbf/CAVIAR/.